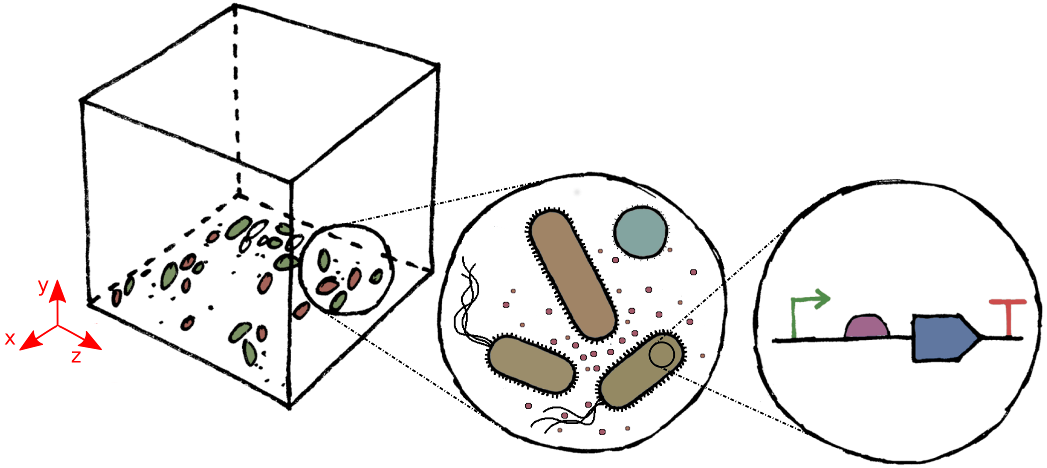
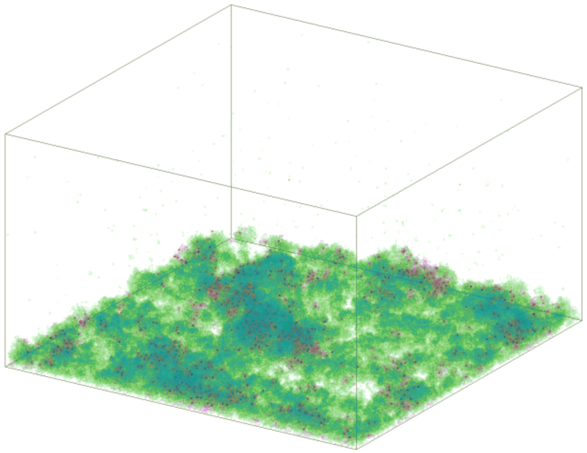
Simbiotics: a multi-scale integrative platform for 3D modelling of bacterial populations

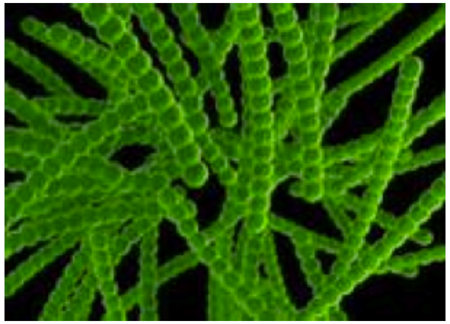




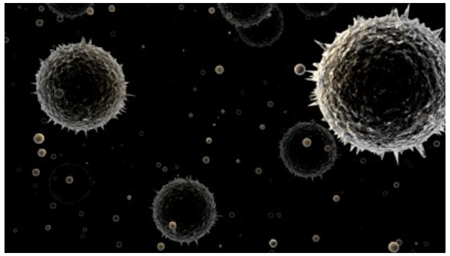
Jonathan Naylor, Dr Harold Fellermann, Prof Natalio Krasnogor

Interdisciplinary Computing and Complex BioSystems (ICOS)
School of Computing Science, Newcastle University

Bacteria Everywhere



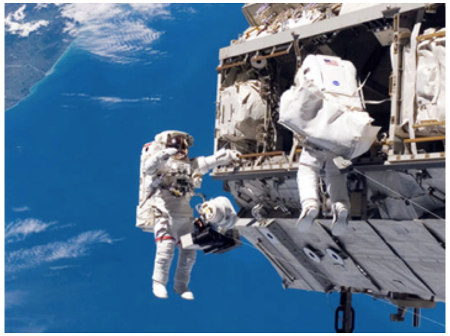
in Water



in Air

b2201212 [RM] © www.visualphotos.com

in Soil



in Space

ASTROPHYSICS
Antimatter-Sparked SUPERNOVAE

NEUROSCIENCE
Building a MACHINE BRAIN

PHYSICS
The Rough Road to FUSION ENERGY

SCIENTIFIC AMERICAN

June 2012

ScientificAmerican.com

Your Inner Ecosystem

In your body, bacteria outnumber your own cells 10 to 1.
Who's in control?

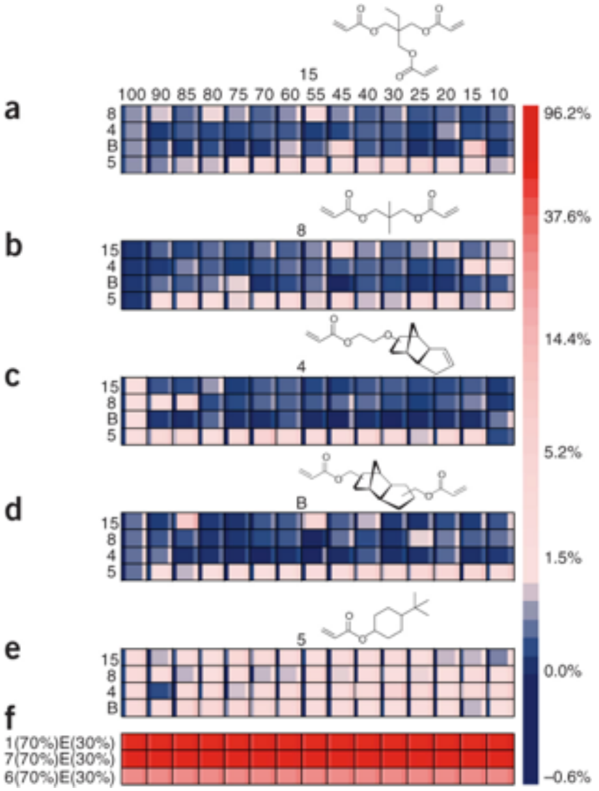
in Animals

in Hospitals

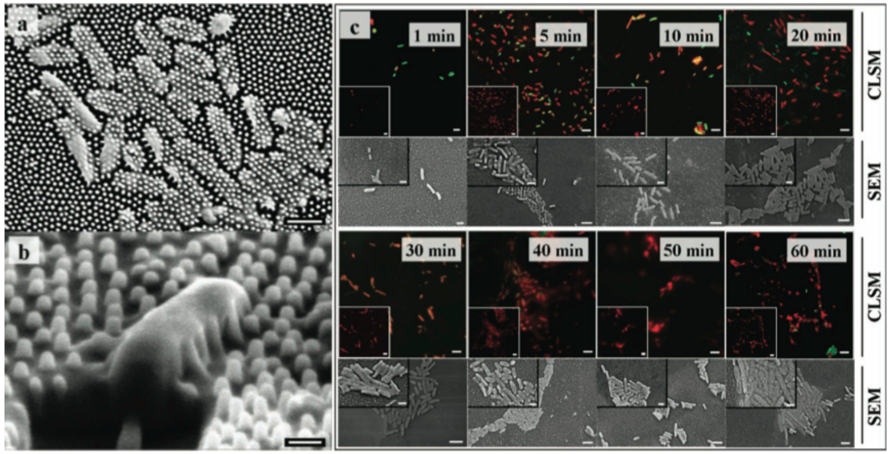
in Factories

in Waterworks

Bacteria Everywhere



Combinatorial discovery of polymers resistant to bacterial attachment. A.L. Hook et al. Nature Biotechnology (2012)



Natural Bactericidal Surfaces: Mechanical Rupture of Pseudomonas aeruginosa Cells by Cicada Wings. E.P. Ivanova et al. Smal 2012.

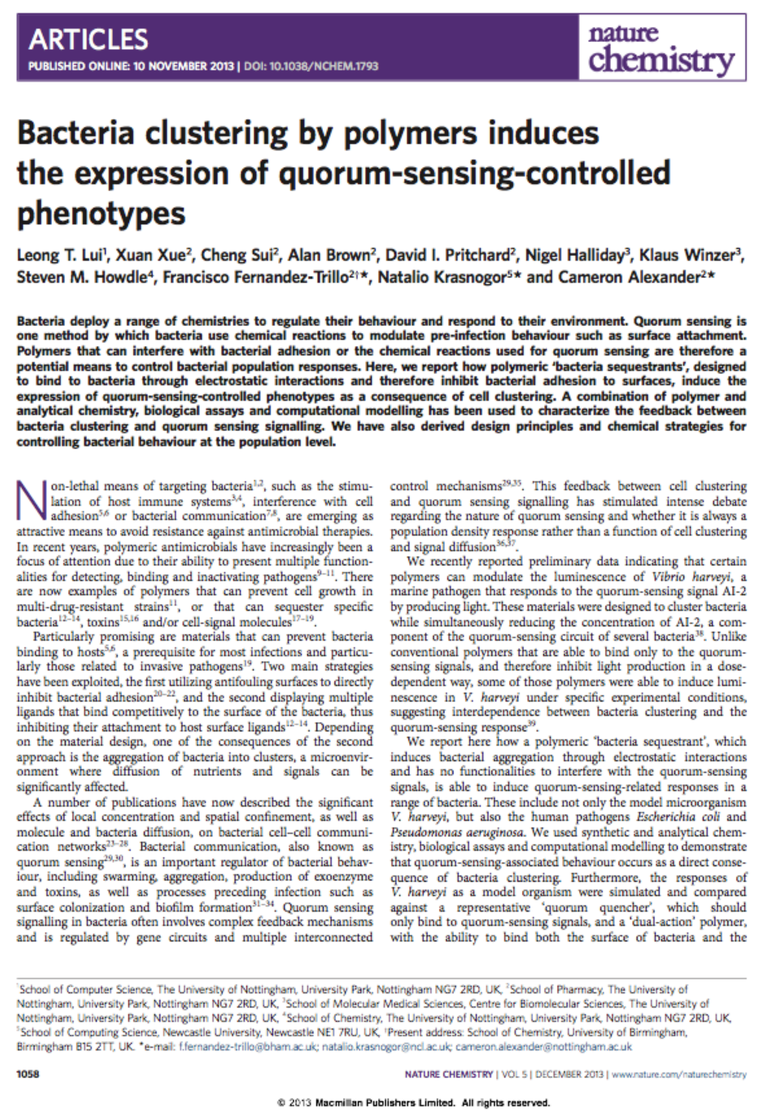
Synthetic Polymers For Controlling QS Dependent Phenotypes

To investigate the effect of clustering in QS phenotype we studied several polymers

Tested in *V. harveyi, E. coli and P. aeruginosa*

We synthetized polymers that were:

- bacterial sequesters
- quorum quenchers
- both



ARTICLES
PUBLISHED ONLINE: 10 NOVEMBER 2013 | DOI: 10.1038/NCHEM.1793
nature chemistry

Bacteria clustering by polymers induces the expression of quorum-sensing-controlled phenotypes

Leong T. Lui[1], Xuan Xue[2], Cheng Sui[2], Alan Brown[2], David I. Pritchard[2], Nigel Halliday[3], Klaus Winzer[3], Steven M. Howdle[4], Francisco Fernandez-Trillo[2†]*, Natalio Krasnogor[5]* and Cameron Alexander[2]*

Bacteria deploy a range of chemistries to regulate their behaviour and respond to their environment. Quorum sensing is one method by which bacteria use chemical reactions to modulate pre-infection behaviour such as surface attachment. Polymers that can interfere with bacterial adhesion or the chemical reactions used for quorum sensing are therefore a potential means to control bacterial population responses. Here, we report how polymeric 'bacteria sequestrants', designed to bind to bacteria through electrostatic interactions and therefore inhibit bacterial adhesion to surfaces, induce the expression of quorum-sensing-controlled phenotypes as a consequence of cell clustering. A combination of polymer and analytical chemistry, biological assays and computational modelling has been used to characterize the feedback between bacteria clustering and quorum sensing signalling. We have also derived design principles and chemical strategies for controlling bacterial behaviour at the population level.

Non-lethal means of targeting bacteria[1,2], such as the stimulation of host immune systems[3,4], interference with cell adhesion[5,6] or bacterial communication[7,8], are emerging as attractive means to avoid resistance against antimicrobial therapies. In recent years, polymeric antimicrobials have increasingly been a focus of attention due to their ability to present multiple functionalities for detecting, binding and inactivating pathogens[9–11]. There are now examples of polymers that can prevent cell growth in multi-drug-resistant strains[11], or that can sequester specific bacteria[12–14], toxins[15,16] and/or cell-signal molecules[17–19].

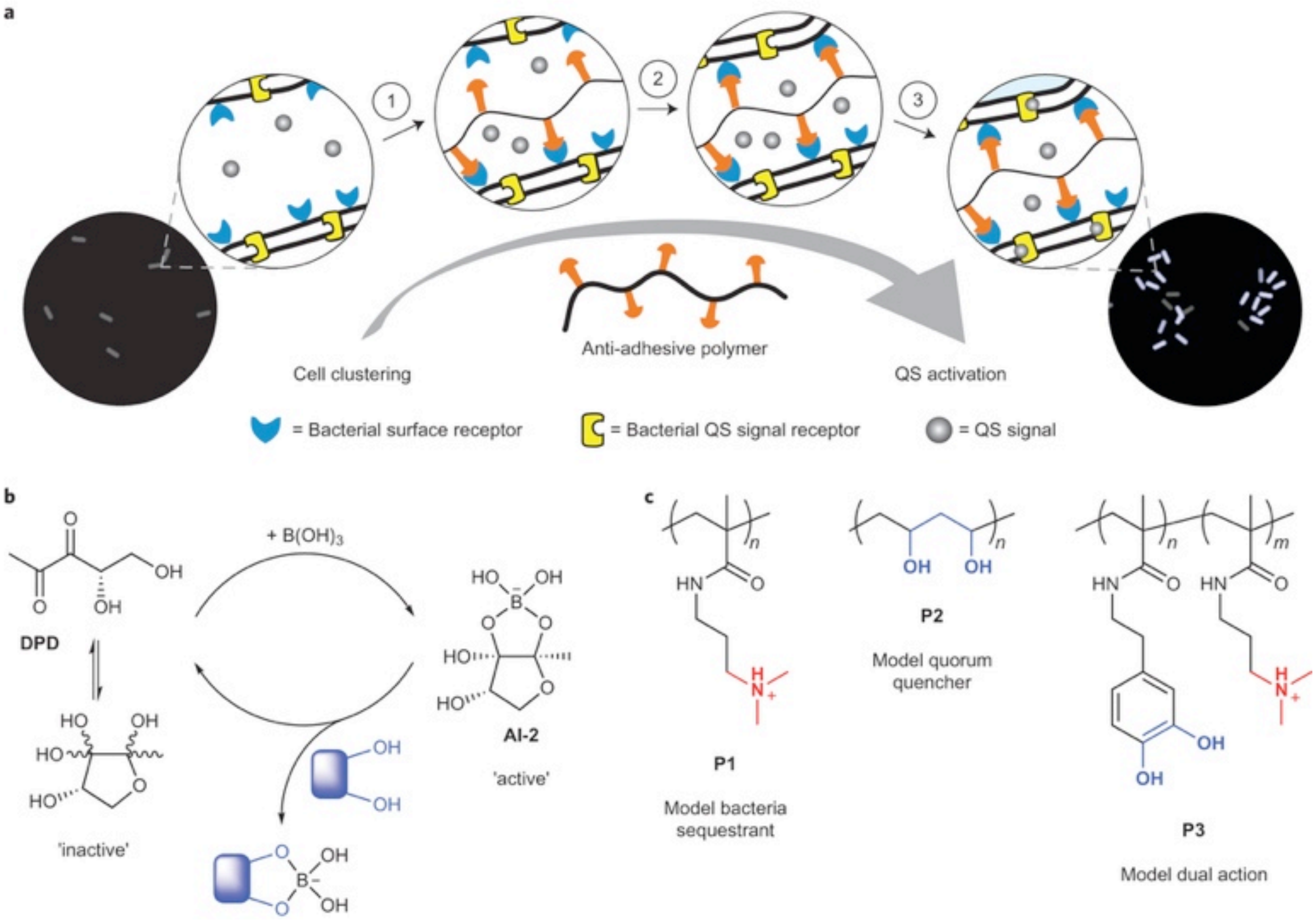
Particularly promising are materials that can prevent bacteria binding to hosts[5,6], a prerequisite for most infections and particularly those related to invasive pathogens[19]. Two main strategies have been exploited, the first utilizing antifouling surfaces to directly inhibit bacterial adhesion[20–22], and the second displaying multiple ligands that bind competitively to the surface of the bacteria, thus inhibiting their attachment to host surface ligands[12–14]. Depending on the material design, one of the consequences of the second approach is the aggregation of bacteria into clusters, a microenvironment where diffusion of nutrients and signals can be significantly affected.

A number of publications have now described the significant effects of local concentration and spatial confinement, as well as molecule and bacteria diffusion, on bacterial cell–cell communication networks[23–28]. Bacterial communication, also known as quorum sensing[29,30], is an important regulator of bacterial behaviour, including swarming, aggregation, production of exoenzyme and toxins, as well as processes preceding infection such as surface colonization and biofilm formation[31–34]. Quorum sensing signalling in bacteria often involves complex feedback mechanisms and is regulated by gene circuits and multiple interconnected control mechanisms[29,35]. This feedback between cell clustering and quorum sensing signalling has stimulated intense debate regarding the nature of quorum sensing and whether it is always a population density response rather than a function of cell clustering and signal diffusion[36,37].

We recently reported preliminary data indicating that certain polymers can modulate the luminescence of *Vibrio harveyi*, a marine pathogen that responds to the quorum-sensing signal AI-2 by producing light. These materials were designed to cluster bacteria while simultaneously reducing the concentration of AI-2, a component of the quorum-sensing circuit of several bacteria[38]. Unlike conventional polymers that are able to bind only to the quorum-sensing signals, and therefore inhibit light production in a dose-dependent way, some of those polymers were able to induce luminescence in *V. harveyi* under specific experimental conditions, suggesting interdependence between bacteria clustering and the quorum-sensing response[39].

We report here how a polymeric 'bacteria sequestrant', which induces bacterial aggregation through electrostatic interactions and has no functionalities to interfere with the quorum-sensing signals, is able to induce quorum-sensing-related responses in a range of bacteria. These include not only the model microorganism *V. harveyi*, but also the human pathogens *Escherichia coli* and *Pseudomonas aeruginosa*. We used synthetic and analytical chemistry, biological assays and computational modelling to demonstrate that quorum-sensing-associated behaviour occurs as a direct consequence of bacteria clustering. Furthermore, the responses of *V. harveyi* as a model organism were simulated and compared against a representative 'quorum quencher', which should only bind to quorum-sensing signals, and a 'dual-action' polymer, with the ability to bind both the surface of bacteria and the

[1] School of Computer Science, The University of Nottingham, University Park, Nottingham NG7 2RD, UK, [2] School of Pharmacy, The University of Nottingham, University Park, Nottingham NG7 2RD, UK, [3] School of Molecular Medical Sciences, Centre for Biomolecular Sciences, The University of Nottingham, University Park, Nottingham NG7 2RD, UK, [4] School of Chemistry, The University of Nottingham, University Park, Nottingham NG7 2RD, UK, [5] School of Computing Science, Newcastle University, Newcastle NE1 7RU, UK, † Present address: School of Chemistry, University of Birmingham, Birmingham B15 2TT, UK. *e-mail: f.fernandez-trillo@bham.ac.uk; natalio.krasnogor@ncl.ac.uk; cameron.alexander@nottingham.ac.uk


a
1
2
3
Anti-adhesive polymer
Cell clustering
QS activation
= Bacterial surface receptor
= Bacterial QS signal receptor
= QS signal
b
DPD
+ B(OH)3
AI-2
'active'
'inactive'
c
P1
Model bacteria sequestrant
P2
Model quorum quencher
P3
Model dual action

The Computational Model (I)

Due to the spatial and time scales of the system, a mesoscopic lattice agent-based approach was used.

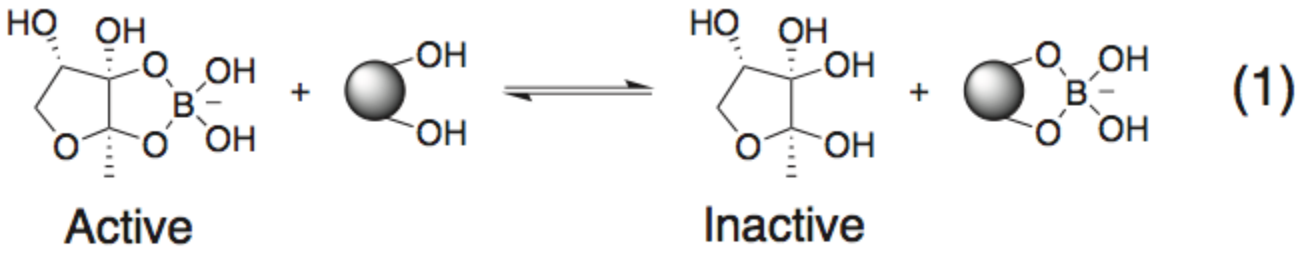
- Three types of object were considered in the model: bacteria (B), polymers (P) and signal molecules (S).

- The size of each B was fixed to occupy a square of 2 × 2 arbitrary lattice spaces. The sizes of S and P were considered to be negligibly small. One unit lattice space could thus contain a quarter of B and unlimited numbers of S and P.

- Two types of change were considered to take place in the system: chemical binding and diffusion of the objects.

The Computational Model (II)

- The mechanism for signal sequestration based on the competitive formation of borate esters is a reversible process as depicted in equation.



- Due to the multivalent nature of the interaction, a very small dissociation constant is expected for polymer binding to bacteria

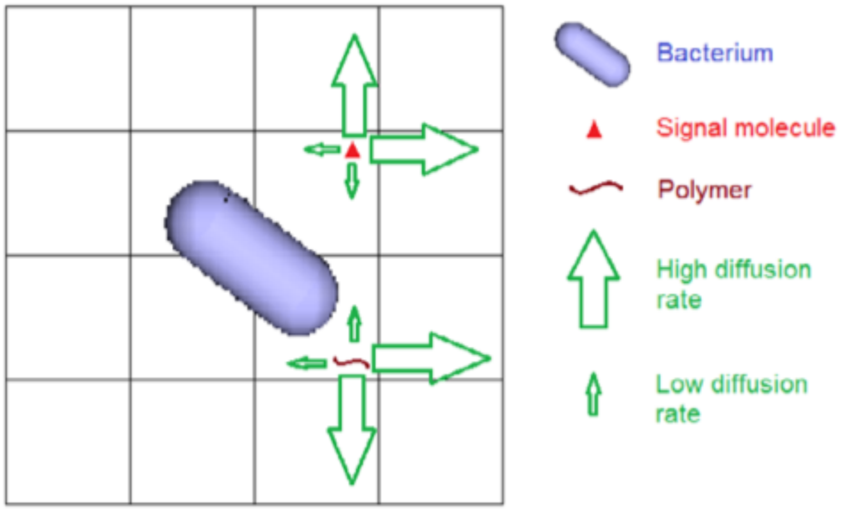
- AI-2 binding sites in the surface of the polymers should be compromised with time as a consequence of a limited accessibility as polymers cluster at the surface of bacteria

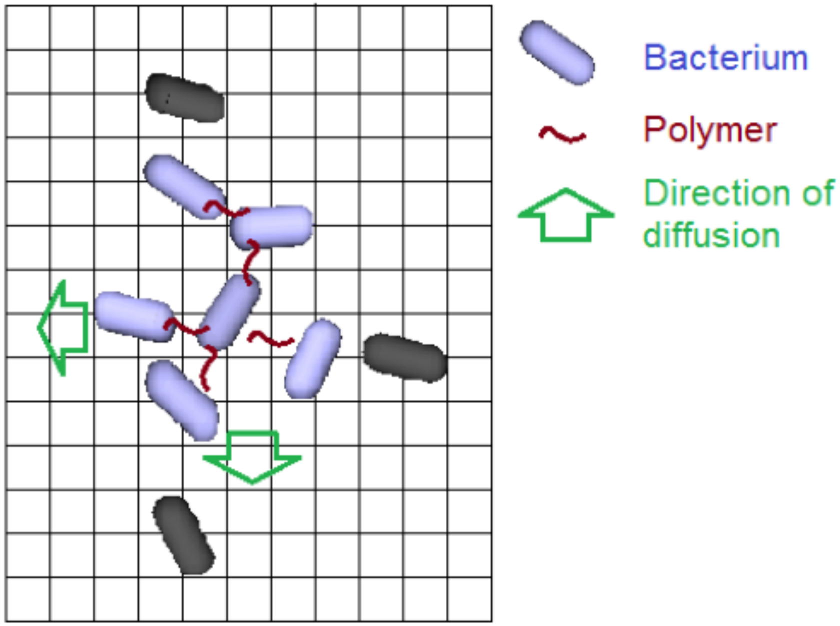
$$i.\quad P_{B,free} + B_{P,free} \underset{}{\overset{K_{PB}}{\rightleftharpoons}} P_B B_P\ ;$$

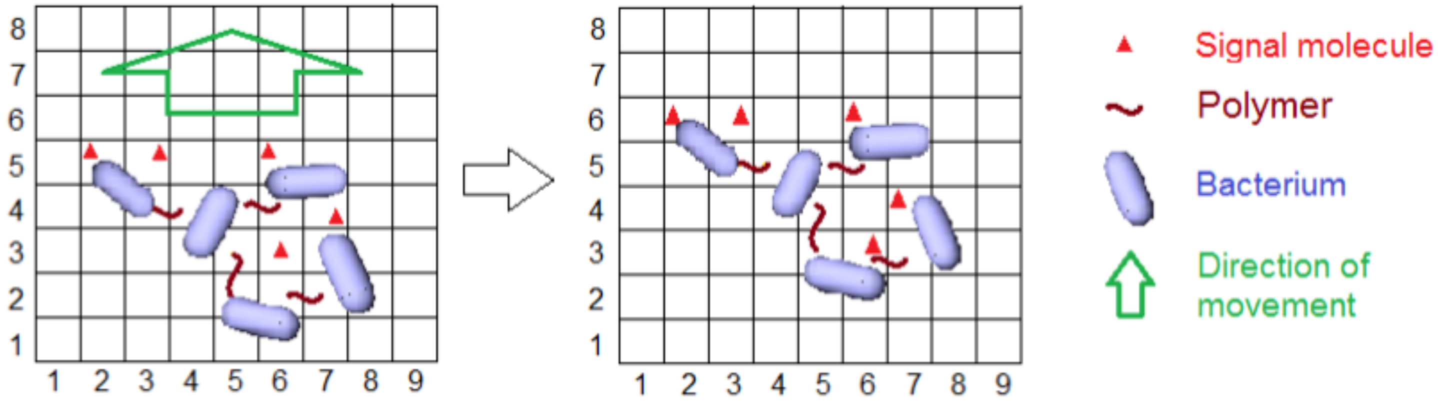
$$ii.\quad B_{S,free} + S \overset{K_{BS}}{\rightleftharpoons} B_S S\ ;$$

$$iii.\quad P_{S,free} + S \overset{K_{PS}}{\rightleftharpoons} P_S S\ ;$$

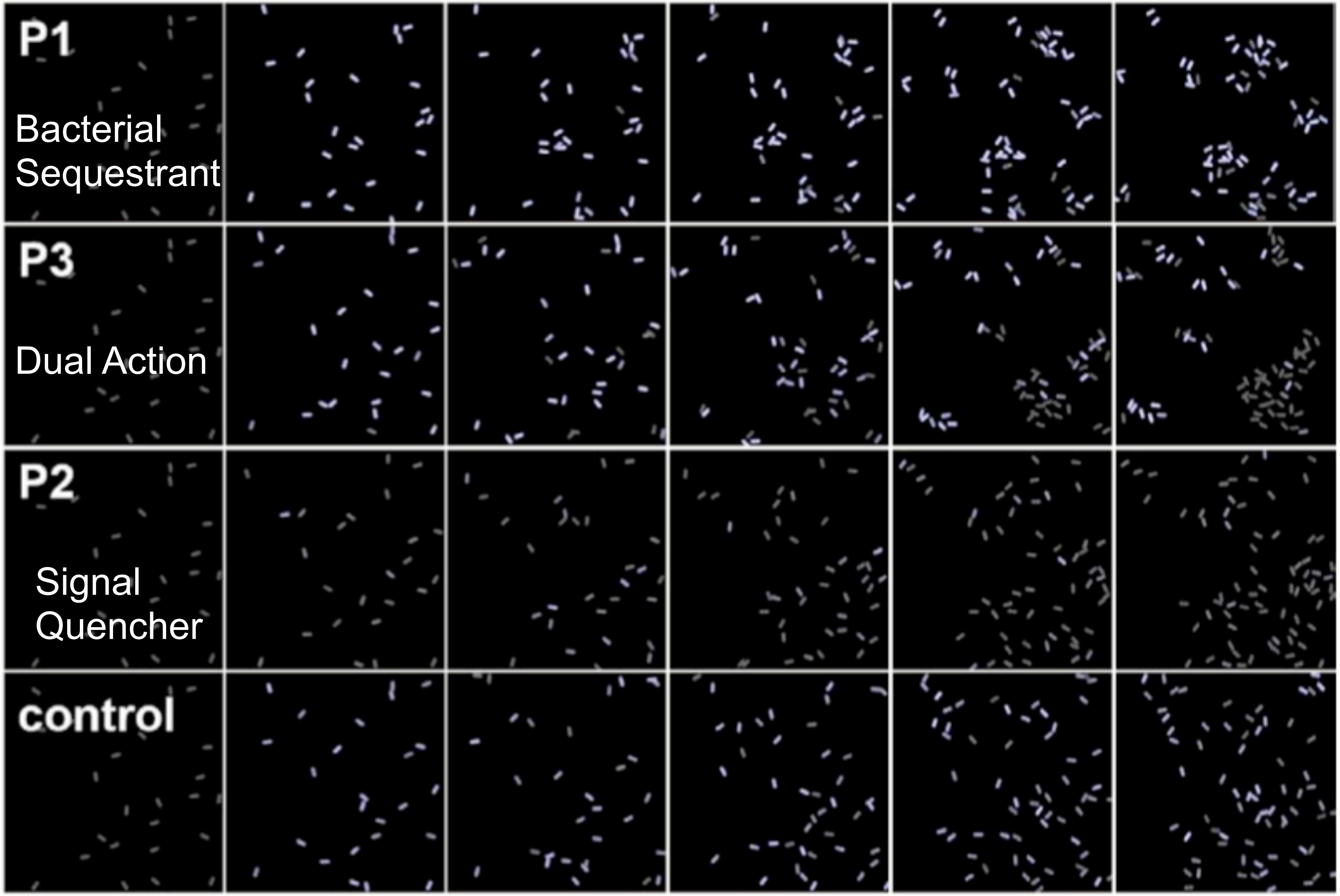
The Computational Model (III)



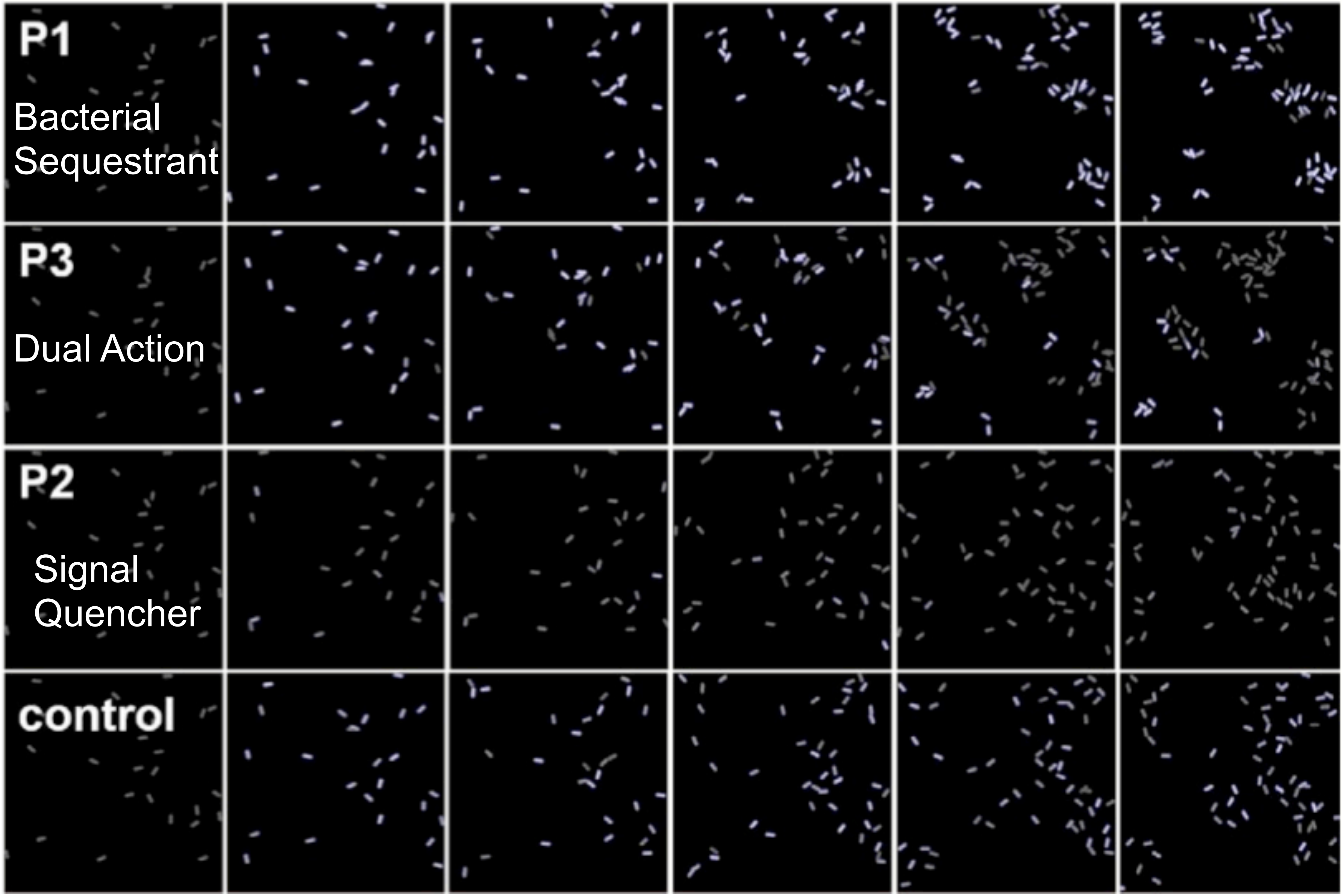




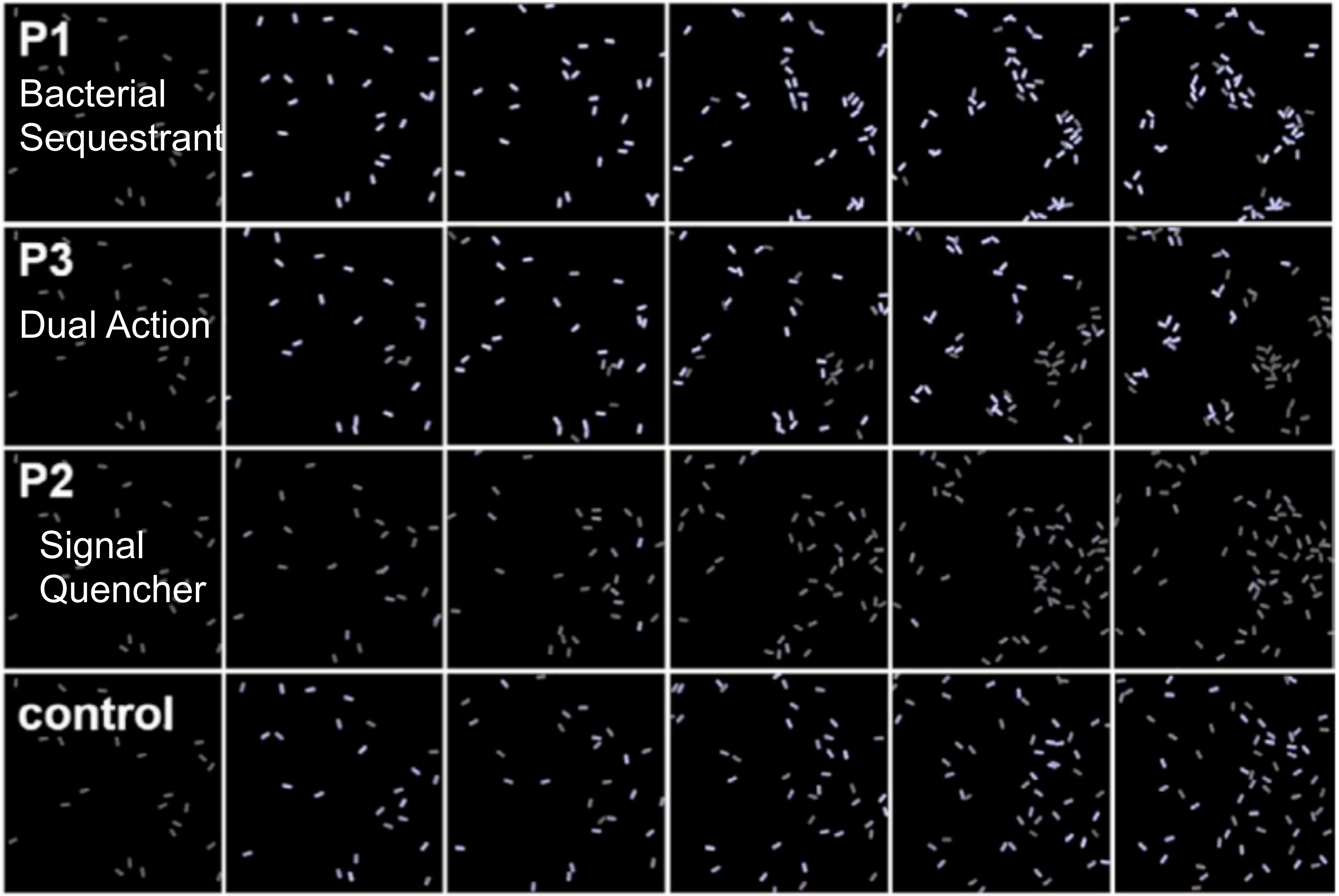
Simulations Results (R1)



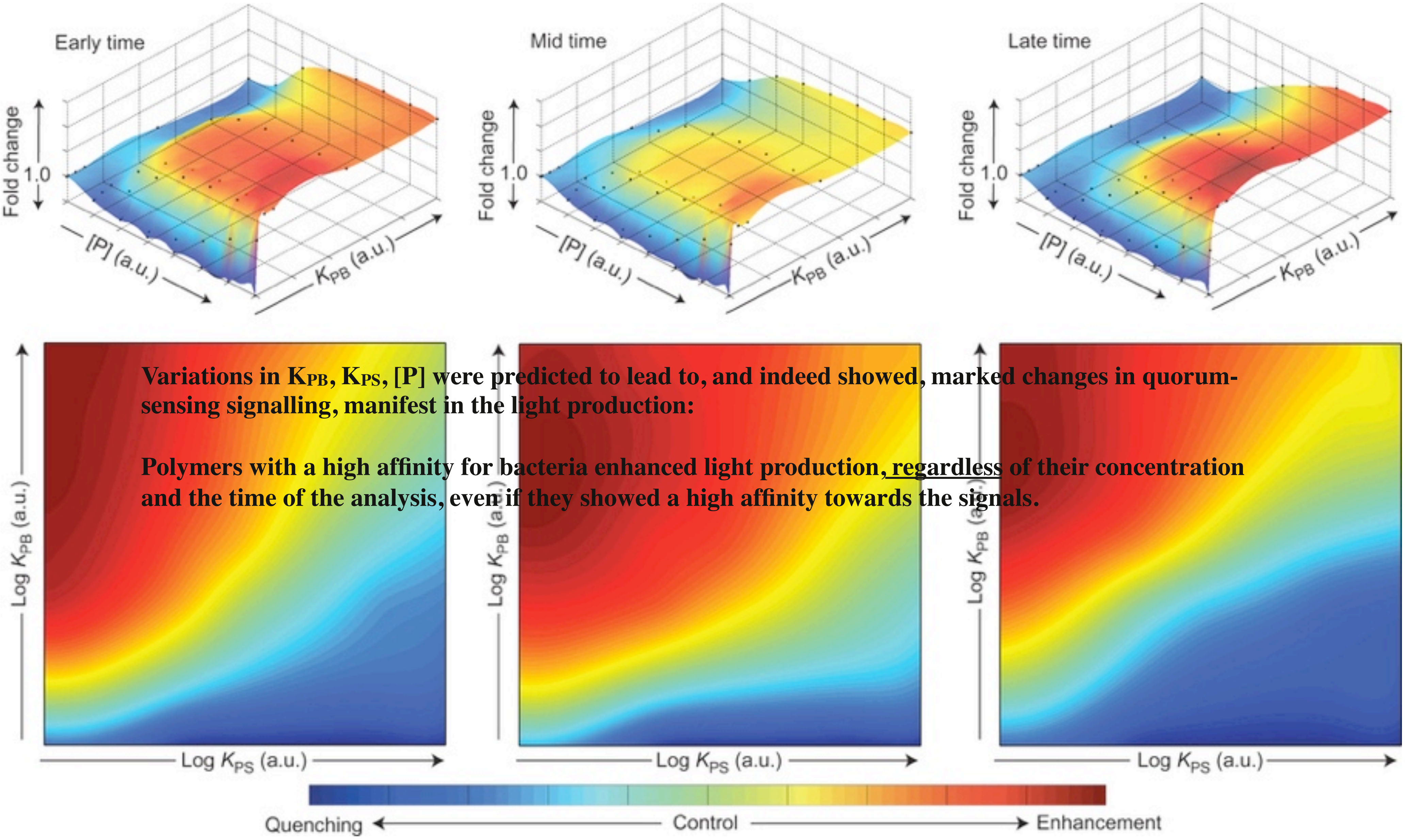
Simulations Results (R2)



Simulations Results (R3)

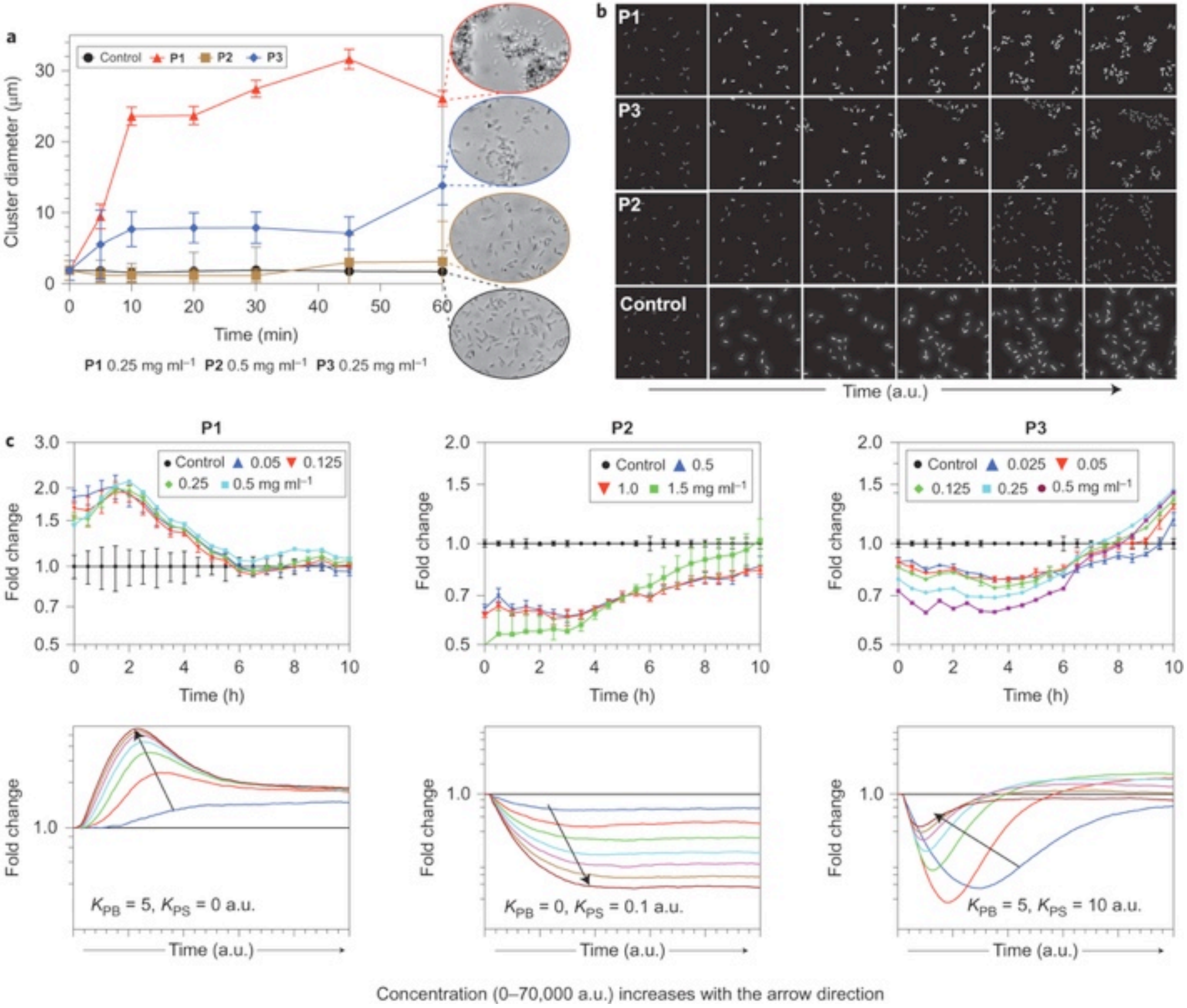


Feedback between the ability of polymers to induce aggregation and interact with signal



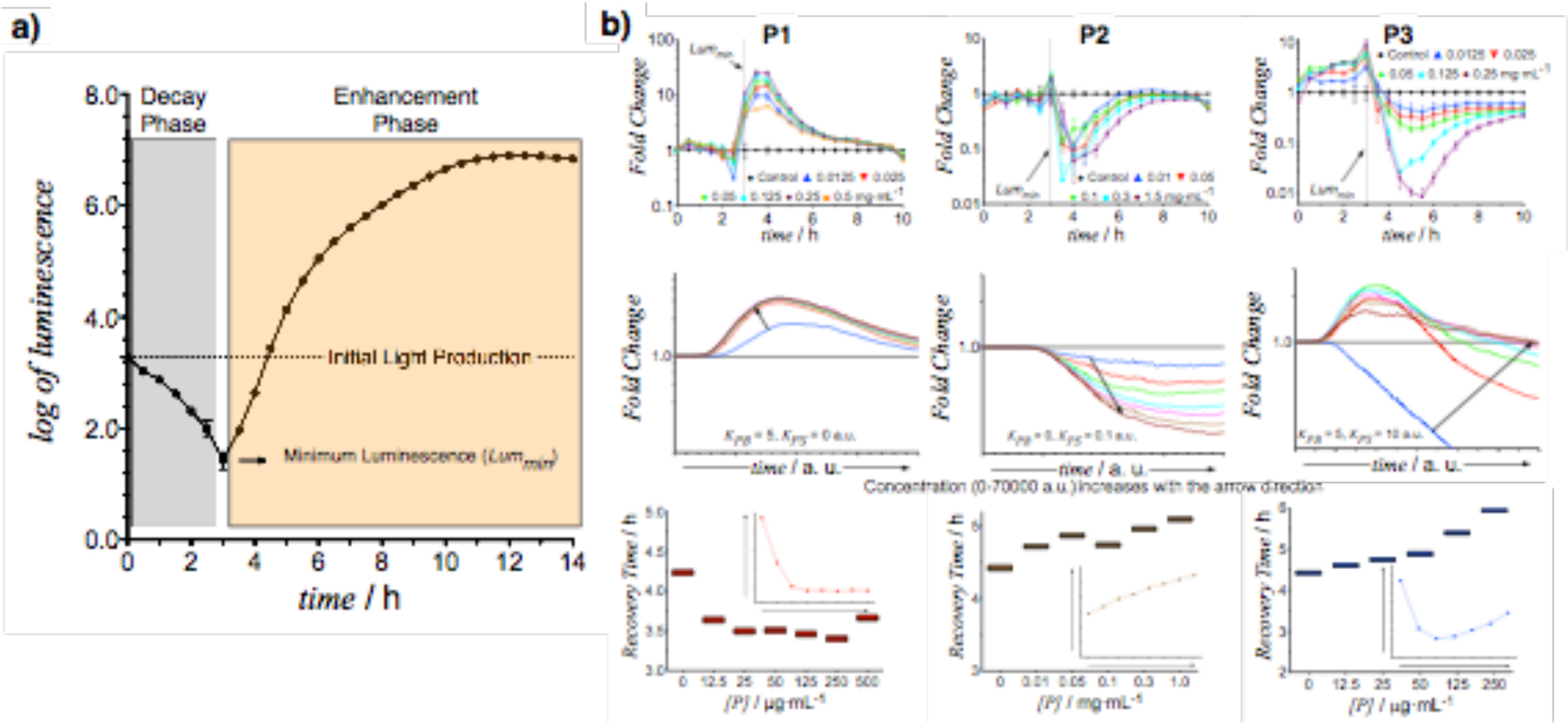
Variations in K_{PB}, K_{PS}, [P] were predicted to lead to, and indeed showed, marked changes in quorum-sensing signalling, manifest in the light production:

Polymers with a high affinity for bacteria enhanced light production, <u>regardless</u> of their concentration and the time of the analysis, even if they showed a high affinity towards the signals.



V. *harveyi* MM32, which responds to exogenous AI-2 but does not produce the quorum-sensing signal precursor 4,5-dihydroxy-2,3-pentanedione (DPD).

V. *harveyi* BB170 strain, which is capable of producing DPD



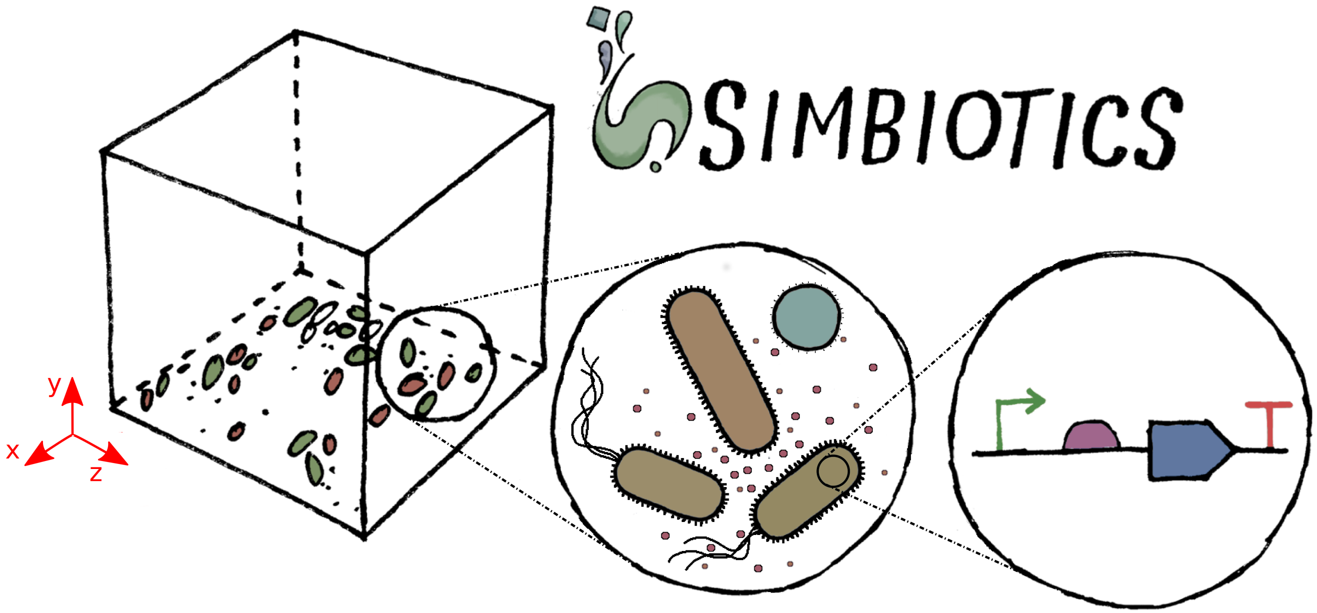
Lessons

Strong Points:

- Abstractions at the right level of details for the questions being posed
- Good match to experiments (phenomenological)
- Not too expensive computationally

Weak Points:

- Not mechanistic model
- HR hungry
- Experienced postdoc working for 2 years on this



- Agent-based modelling platform for multicellular systems, focusing on bacterial populations such as biofilms

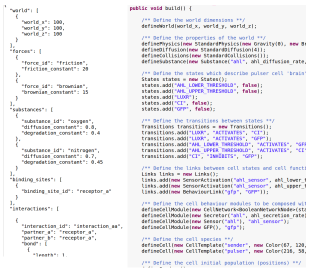
- Simulating a spatially explicit physical world with chemical and biological processes

- Allows modelling of feedback between cell genetics and population behaviour

- Runs on laptop but can scale to millions of cells on HPC (multi-threaded & multi-CPU)

- Flexible modelling environment ***for rapid prototyping***

Design

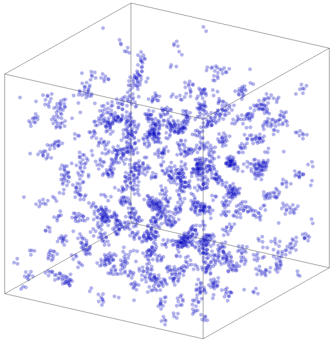


```
{
  "world": [
    {
      "world_x": 100,
      "world_y": 100,
      "world_z": 100
    }
  ],
  "forces": [
    {
      "force_id": "friction",
      "friction_constant": 20
    },
    {
      "force_id": "brownian",
      "brownian_constant": 15
    }
  ],
  "substances": [
    {
      "substance_id": "oxygen",
      "diffusion_constant": 0.8,
      "degradation_constant": 0.4
    },
    {
      "substance_id": "nitrogen",
      "diffusion_constant": 0.7,
      "degradation_constant": 0.45
    }
  ],
  "binding_sites": [
    {
      "binding_site_id": "receptor_a"
    }
  ],
  "interactions": [

    {
      "interaction_id": "interaction_aa",
      "partner_a": "receptor_a",
      "partner_b": "receptor_a",
      "bond": [
        {
          "length": 1,
```

```
public void build() {

    /** Define the world dimensions **/
    defineWorld(world_x, world_y, world_z);

    /** Define the properties of the world **/
    definePhysics(new StandardPhysics(new Gravity(0), new Br
    defineDiffusion(new StandardDiffusion(4));
    defineCollisions(new StandardCollisions());
    defineSubstance(new Substance("ahl", ahl_diffusion_rate,

    /** Define the states which describe pulser cell 'brain'
    States states = new States();
    states.add("AHL_LOWER_THRESHOLD", false);
    states.add("AHL_UPPER_THRESHOLD", false);
    states.add("LUXR");
    states.add("CI", false);
    states.add("GFP", false);

    /** Define the transitions between states **/
    Transitions transitions = new Transitions();
    transitions.add("LUXR", "ACTIVATES", "CI");
    transitions.add("LUXR", "ACTIVATES", "GFP");
    transitions.add("AHL_LOWER_THRESHOLD", "ACTIVATES", "GFP
    transitions.add("AHL_UPPER_THRESHOLD", "ACTIVATES", "CI'
    transitions.add("CI", "INHIBITS", "GFP");

    /** Define the links between cell states and cell functi
    Links links = new Links();
    links.add(new SensorActivation("ahl_sensor", ahl_lower_t
    links.add(new SensorActivation("ahl_sensor", ahl_upper_t
    links.add(new BehaviourLink("gfp", "GFP"));

    /** Define the cell behaviour modules to be composed wit
    defineCellModule(new CellNetwork<BooleanNetworkNode>(sta
    defineCellModule(new Secretor("ahl", ahl_secretion_rate)
    defineCellModule(new Sensor("ahl"), "ahl_sensor");
    defineCellModule(new GFP(), "gfp");

    /** Define the cell species **/
    defineCell(new CellTemplate("sender", new Color(67, 120,
    defineCell(new CellTemplate("pulser", new Color(216, 58,

    /** Define the cell initial population (positions) **/
```

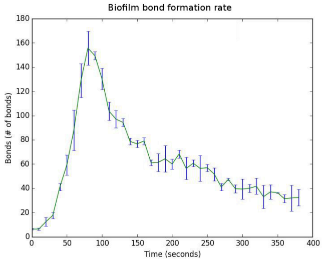
Specify species behaviour
Set positions of cells
Set environmental properties

Simulation



3D physical simulations
Multi-thread and Multi-CPU

Analysis



Probe model properties
Run virtual lab machines
Take pictures/video

Simbiotics Features

Modelling

Physics

- 3D geometries, forces
- Boundary conditions

Chemistry

- diffusion
- reactions, degradation

Biology (cell processses/behaviours)

- Motility (run & tumble, chemotaxis)
- Metabolism, gene regulation
- Membrane transport
- Surface-mediated interactions (receptor-adhesin)
- Conjugation

Expressing cell dynamics

- Ordinary differential equations, Gillespie stochastic simulations, boolean networks, SBML submodels
- Conditional actions (eg. *cell_age > T ? grow_flagellar*)

Simbiotics Features

Analysis

Statistical physics

- Mean-squared displacement
- Velocity autocorrelation function

Virtual lab

- Virtual devices for model interaction (pipette, microsensor)
- Spectrophotometer
- Population maps
- Cell and lineage tracking
- Live graph plotting, custom visualisations etc.
- Customizable data collection and analysis

Simbiotics Features

Workflow Integration and tool placement

Integration

- Microscopy image processing

a

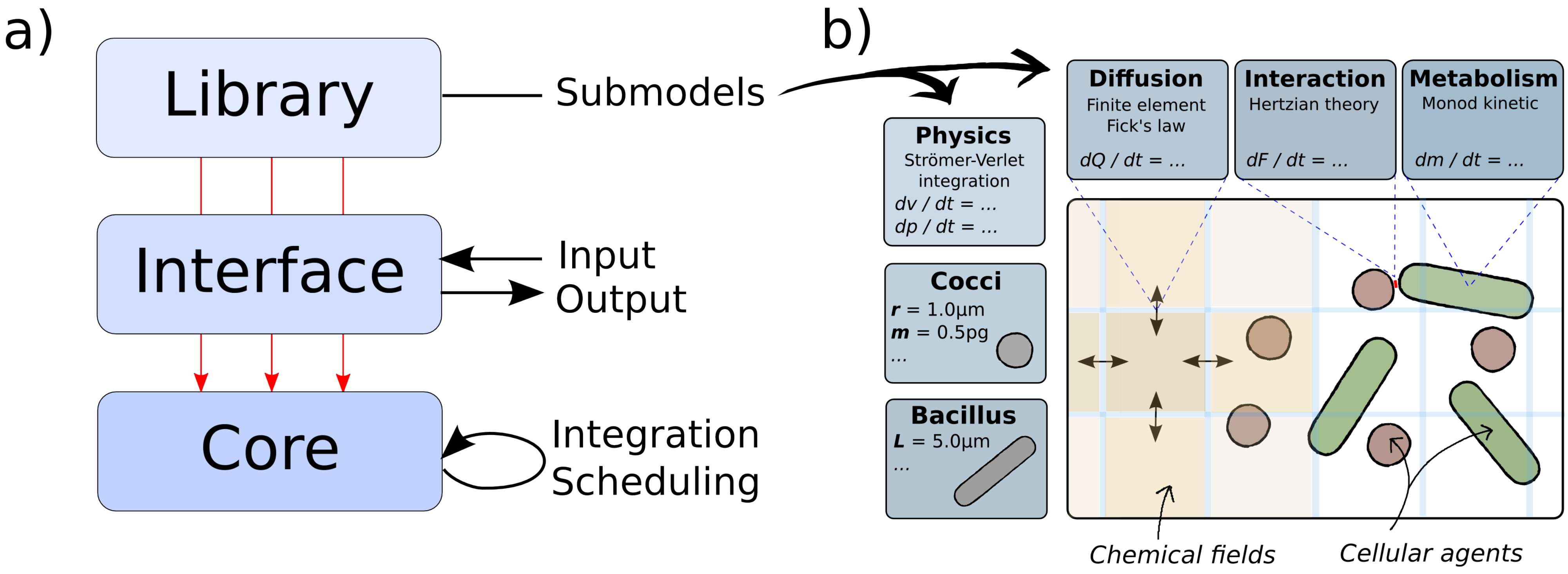
- SBML sub models (can have a unique one for each cell)
- Parameter sweeps (like well-plates)

Data exporters

a

- Export in user-defined formats, such as .csv, .pov (for post-rendering in POVRay
- Live graphs or easy to plot data files for easy knowledge extraction

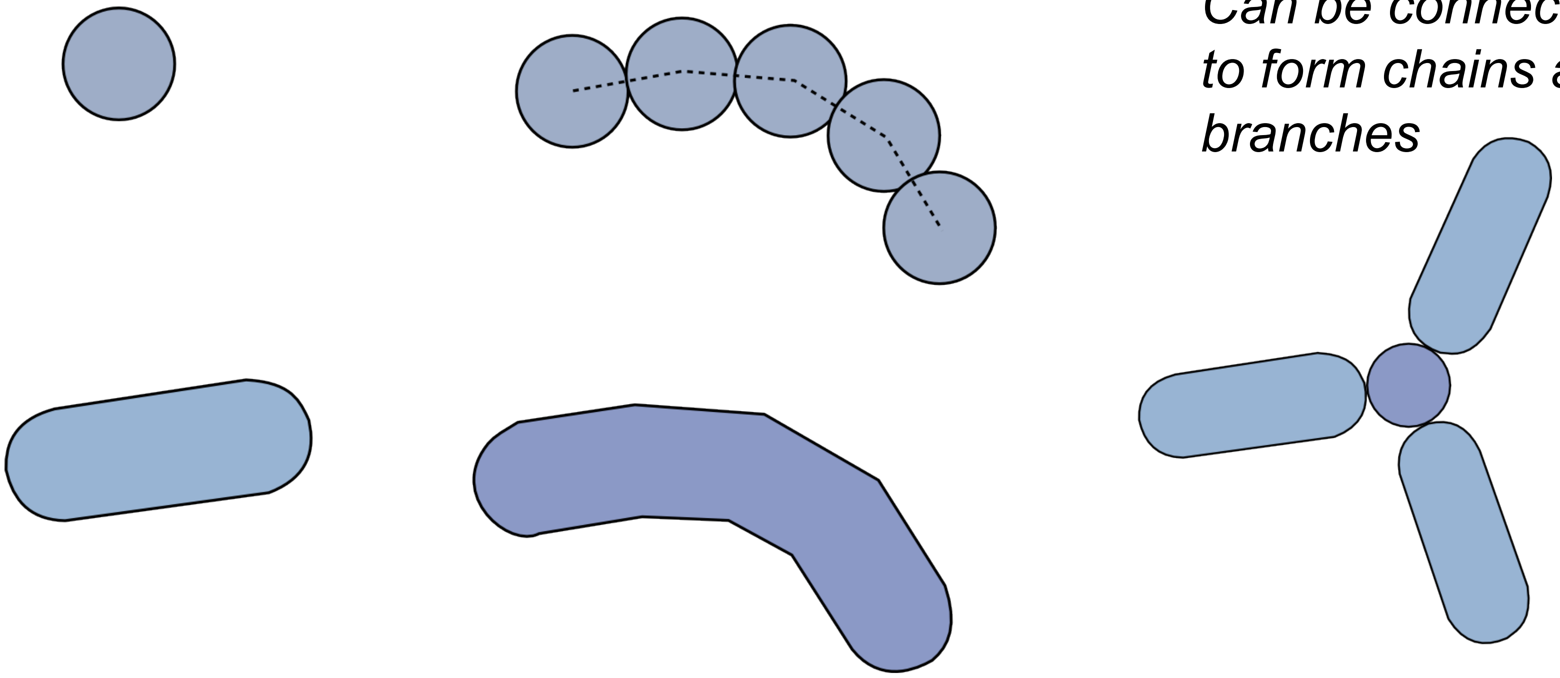
Modular platform architecture and model design



Complex geometry modelling

User defined objects – constructed by composition of spheres, connected by springs (mass-spring system)

Each connection may either be a 'segment' or a 'tube'.



Segmented:

Tubular:

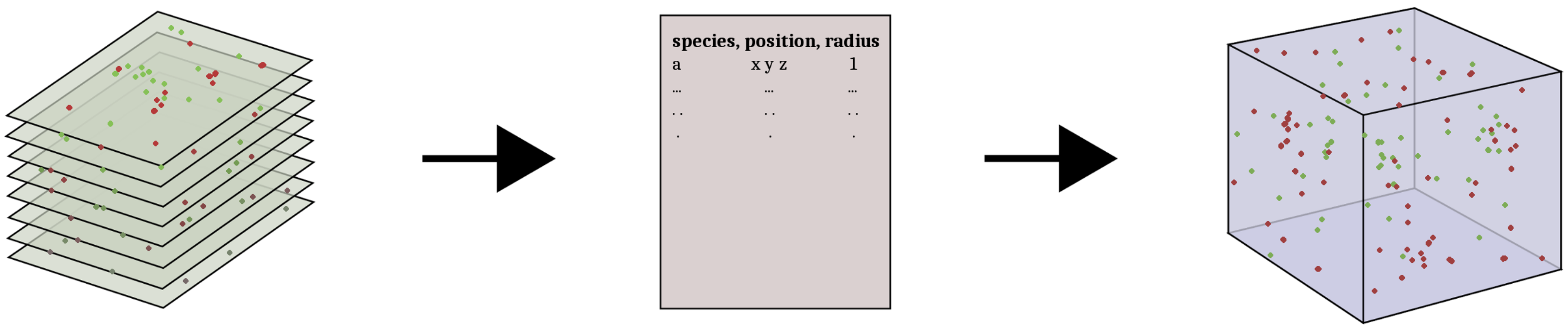
Can be connected to form chains and branches

Model initialisation from microscopy

Dr Yuchun Ding, ICOS, School of Computing Science, Newcastle University

a)

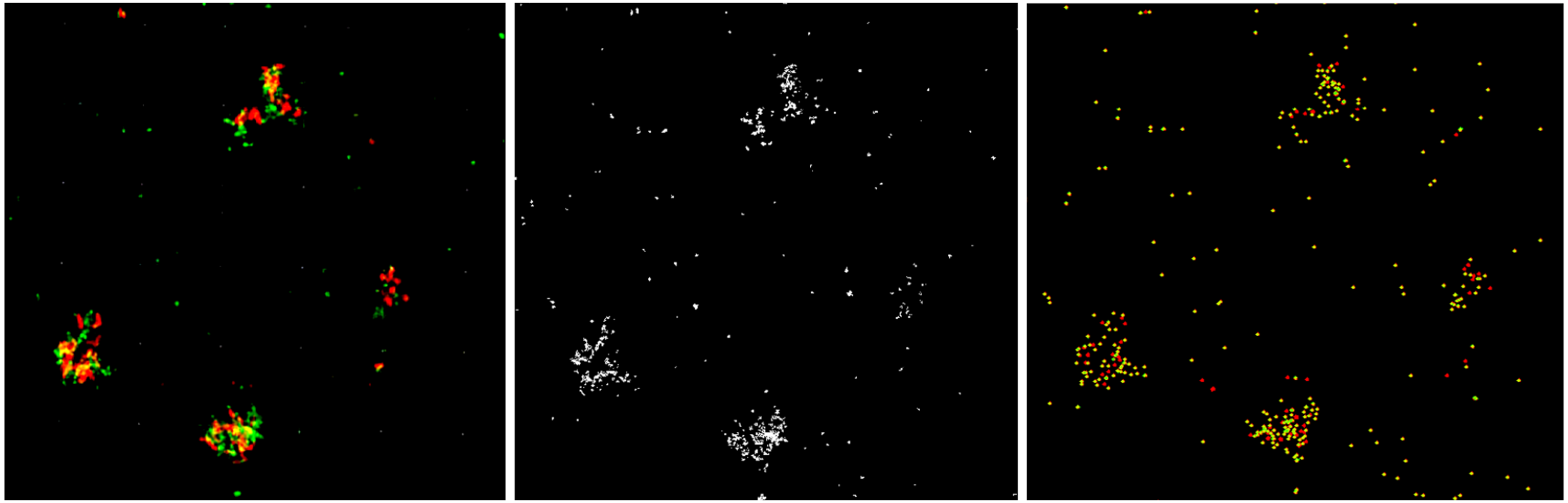
Microscopy z-stack **Z-stack encoding** **Simbiotics state**



species, position, radius
a xyz 1
... ...

Image processing *Model initialisation*

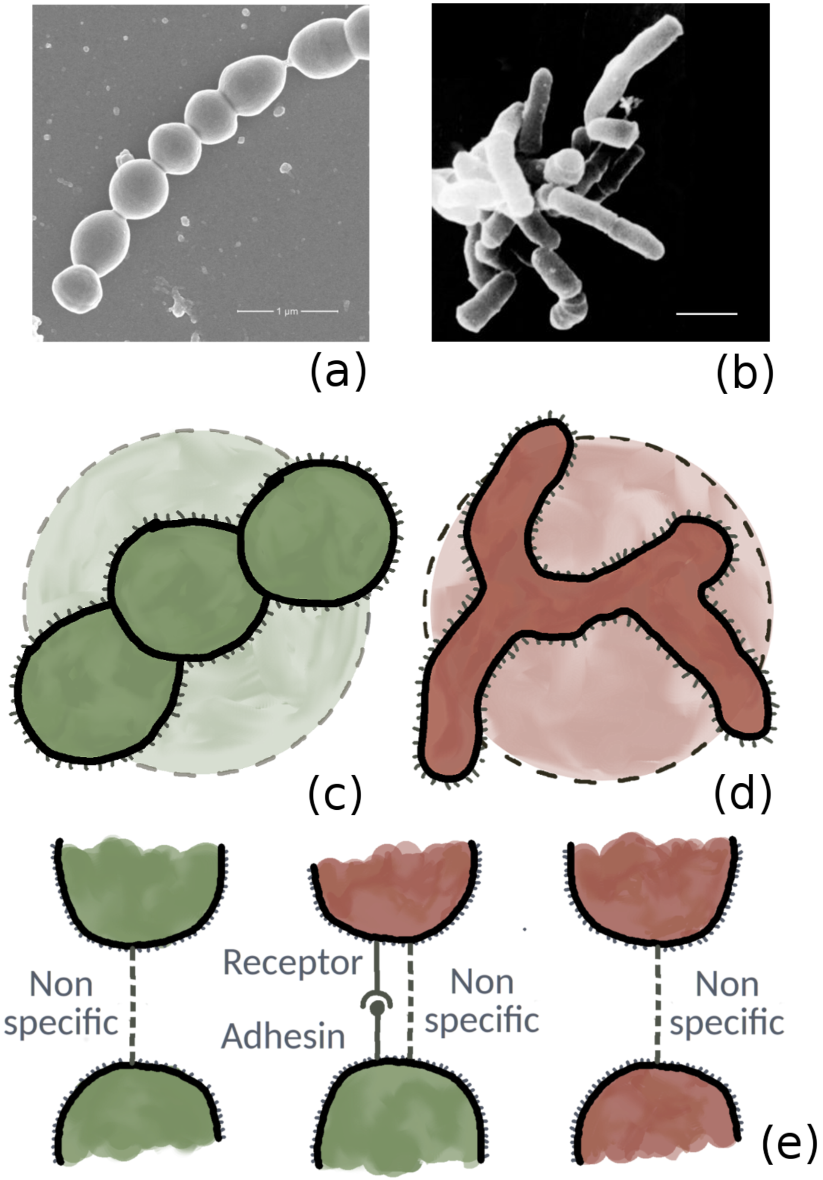
b)



Case Study: Bacterial aggregation

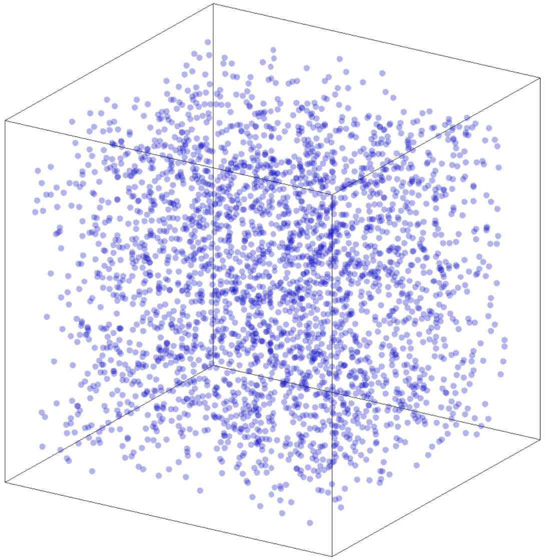
In collaboration with:
Waleed Mohammed and Dr Nick Jakubovics, School of Dental Sciences, Newcastle University

Streptococcus gordonii

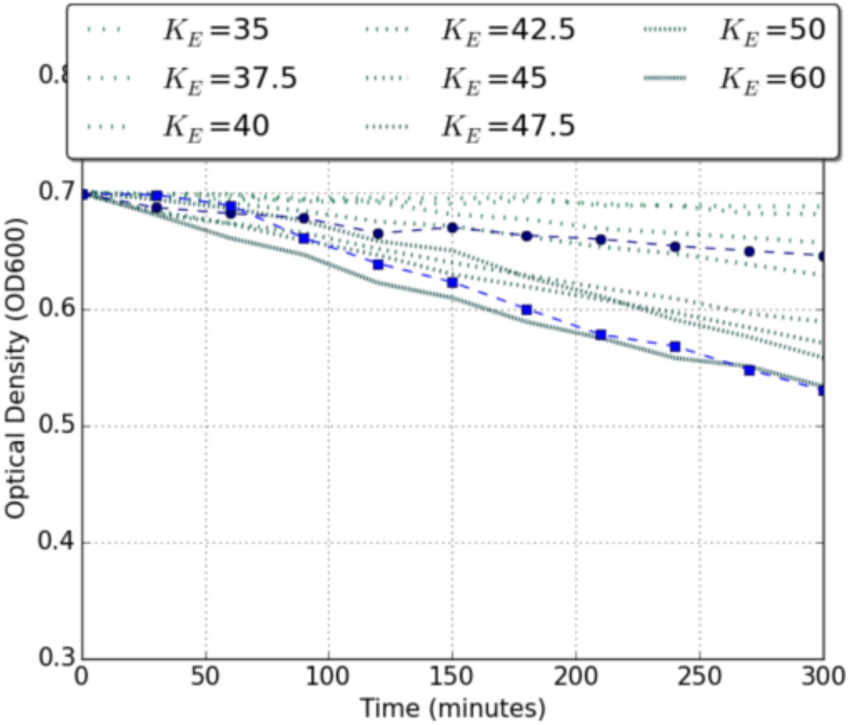


Actinomyces oris

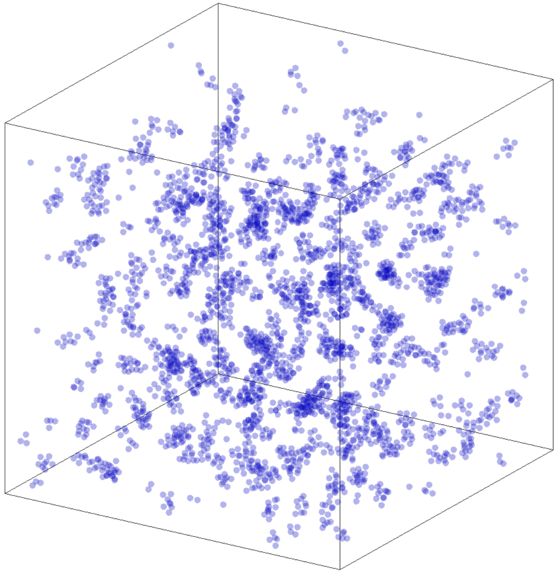
Case Study: Bacterial aggregation



initially well mixed population

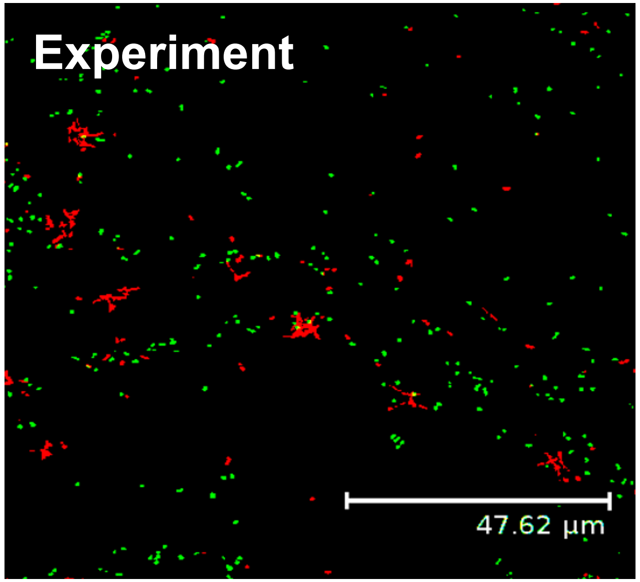


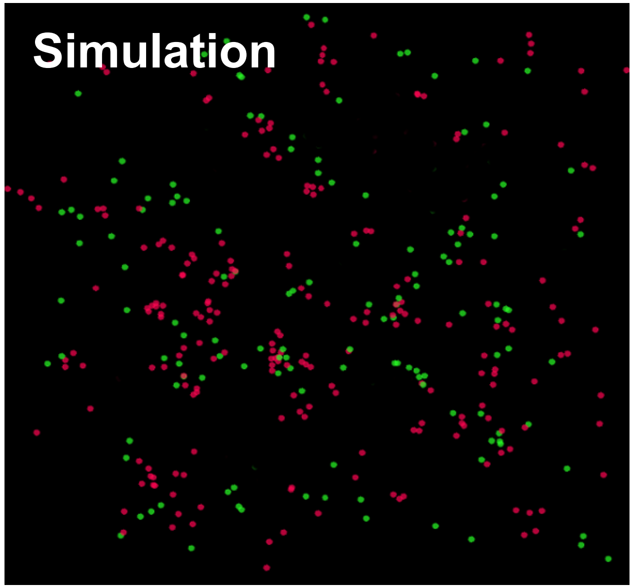
experimental and simulated optical density measurements



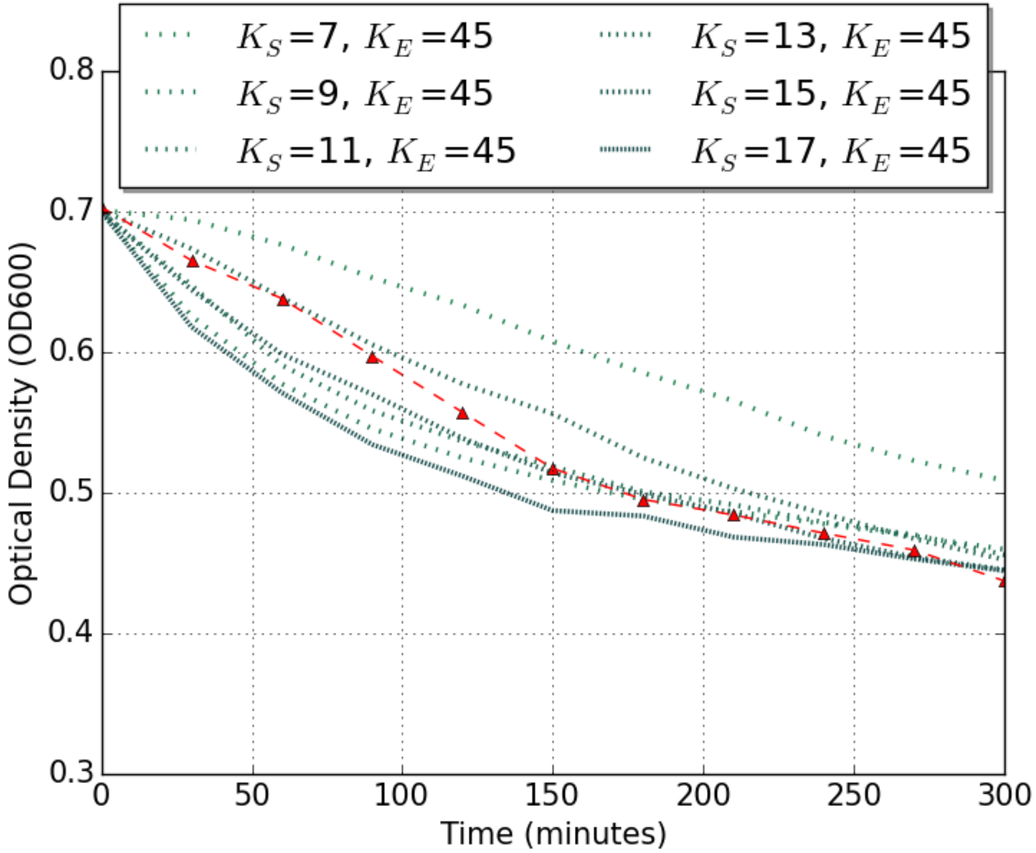
formation of aggregates

Case Study: Bacterial co-aggregation





experimental and simulated optical density measurements



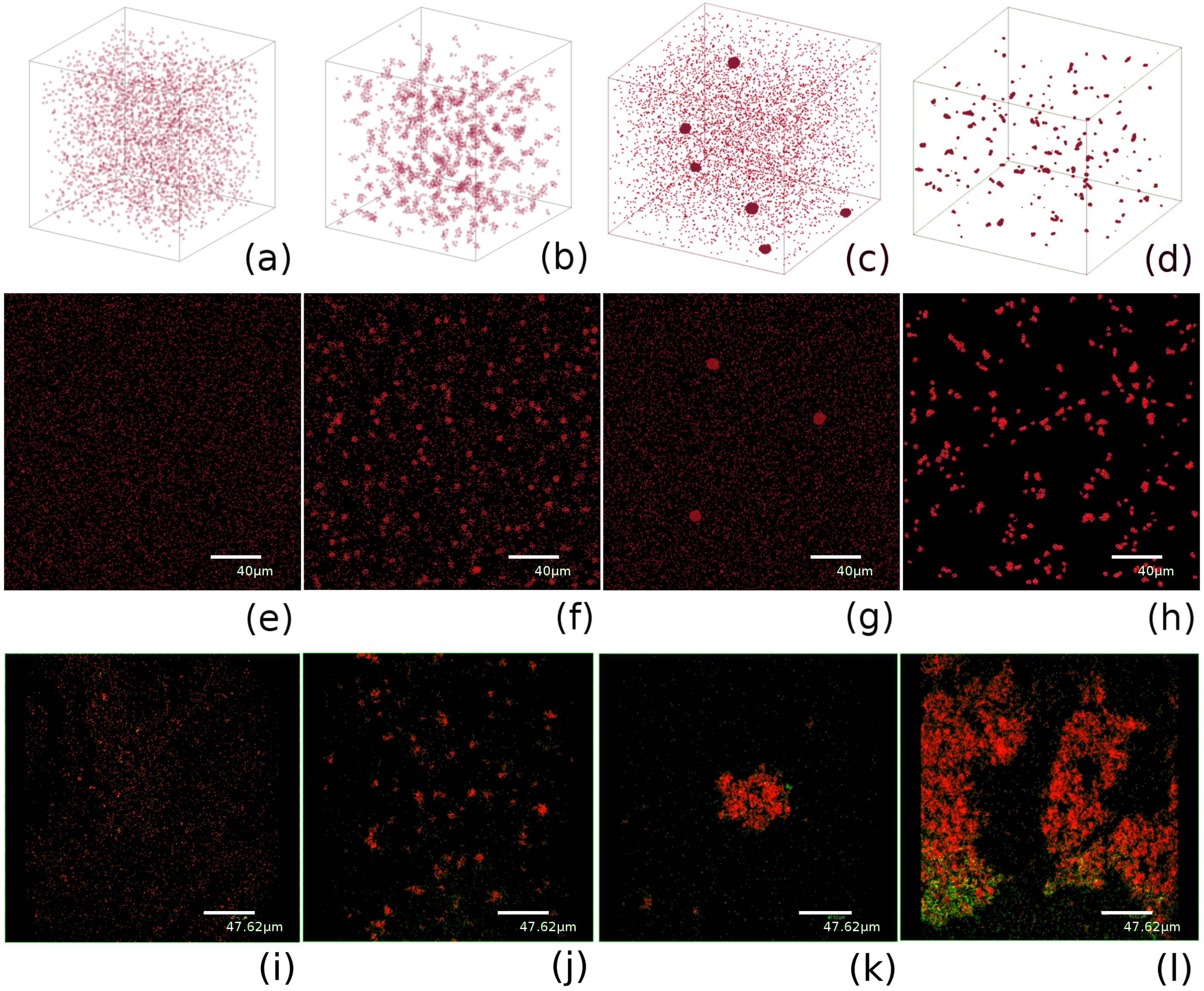
Characterising the parameter space

F = *Brownian motion force*

P = *Probability of interaction for colliding cells*

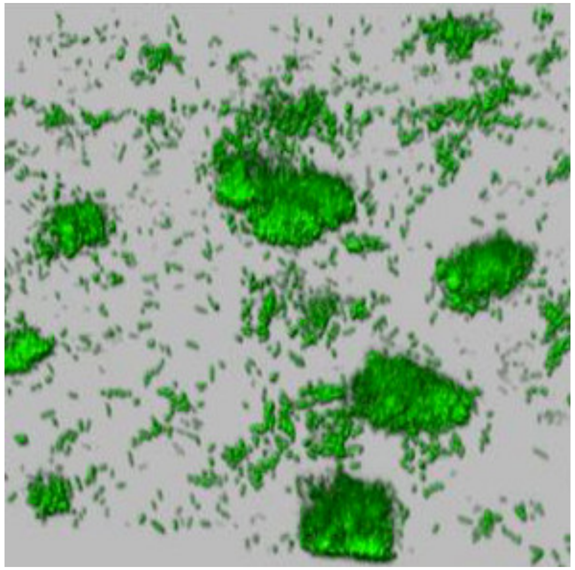
K = *Receptor-adhesin interaction force*

Case Study: Bacterial co-aggregation



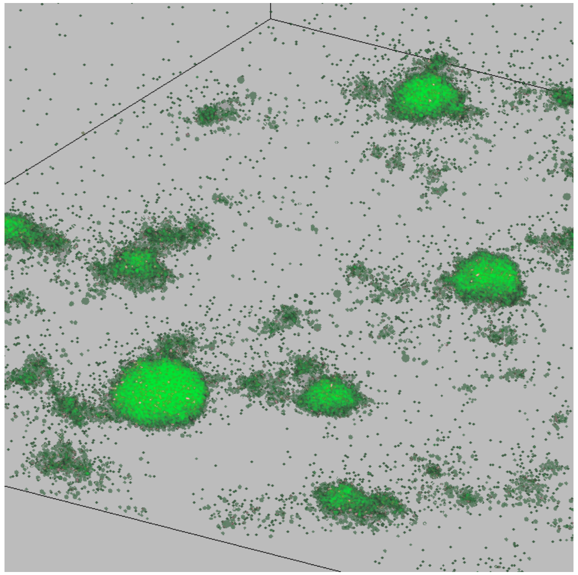
Case Study: Biofilm formation

In collaboration with:
Joy Mukherjee and Catherine Biggs, Department of Chemical Engineering, University of Sheffield
Phillip Wright, Agriculture and Engineering, Newcastle University

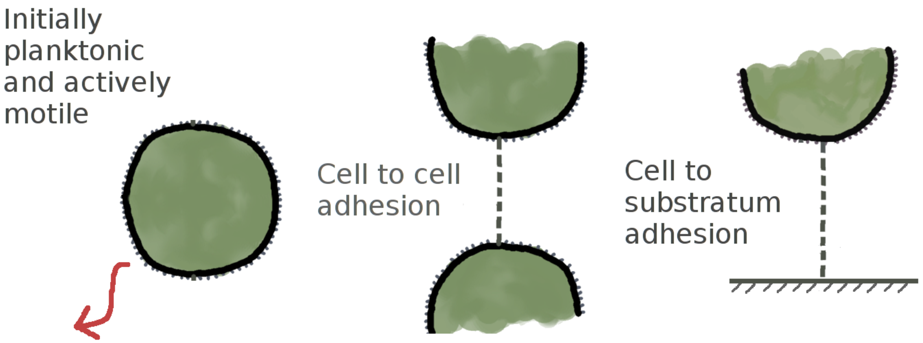


Microscopy image

Synthetic *E. coli* forming biofilm



Simbiotics simulation

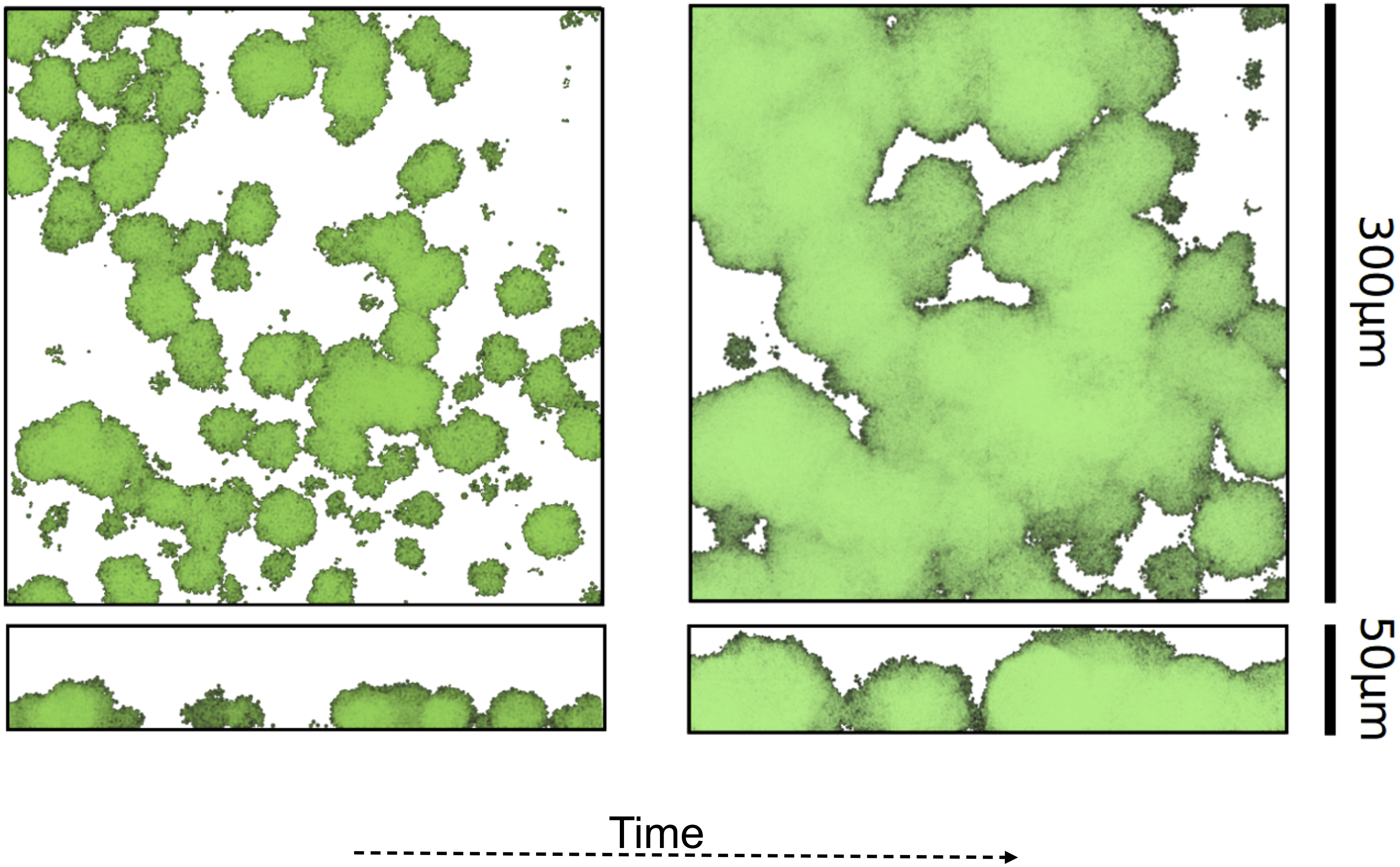


Initially planktonic and actively motile

Cell to cell adhesion

Cell to substratum adhesion

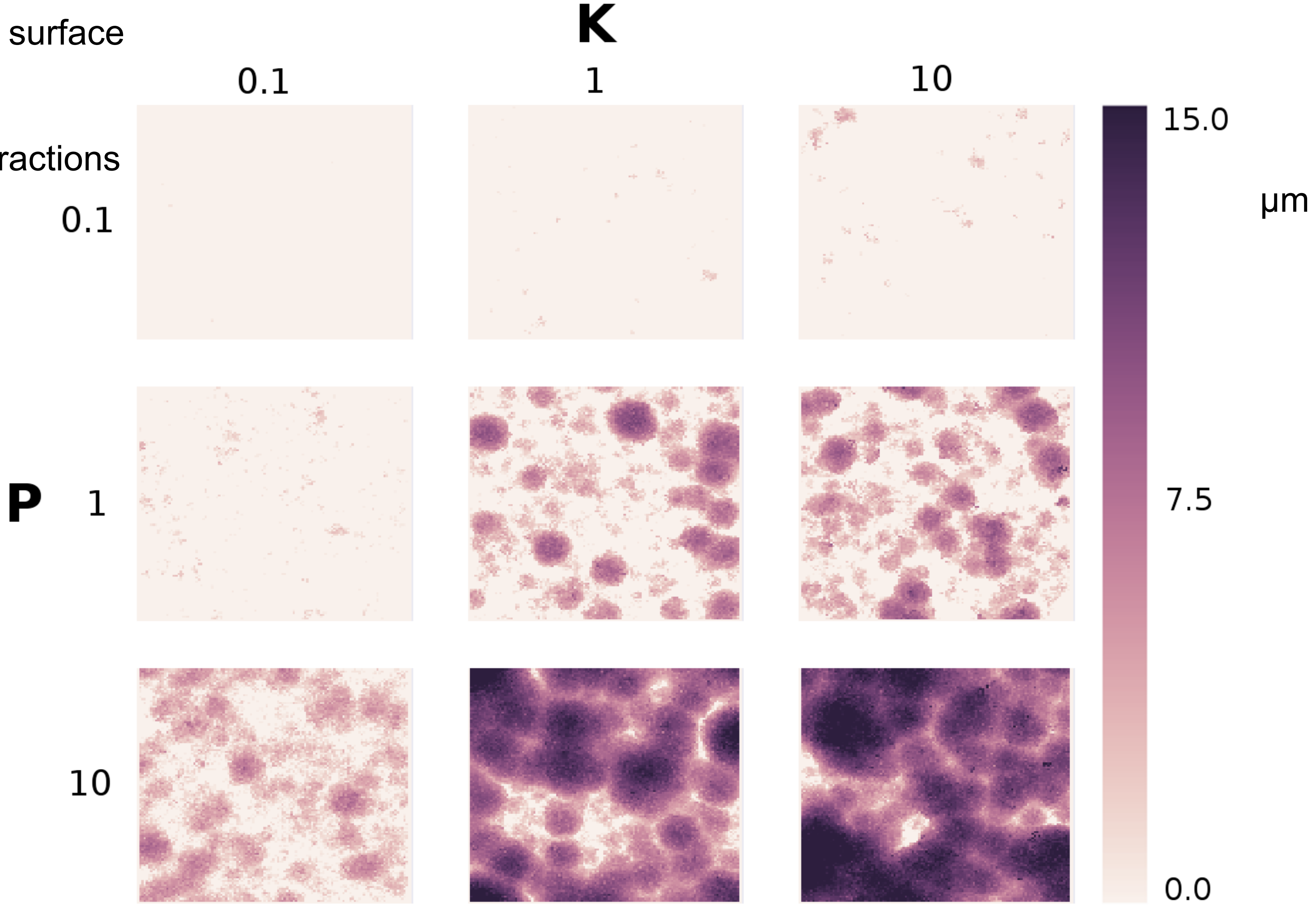
Case Study: Biofilm formation



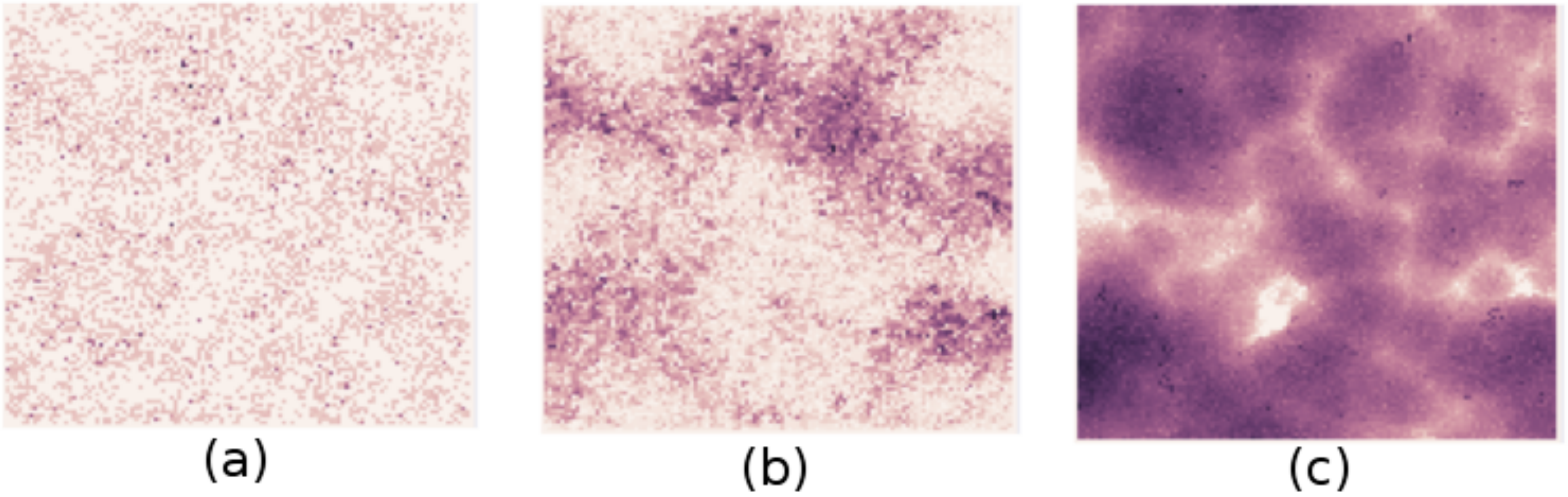
Case Study: Biofilm formation

K = strength of surface interaction

P = rate of interactions



Case Study: Biofilm formation

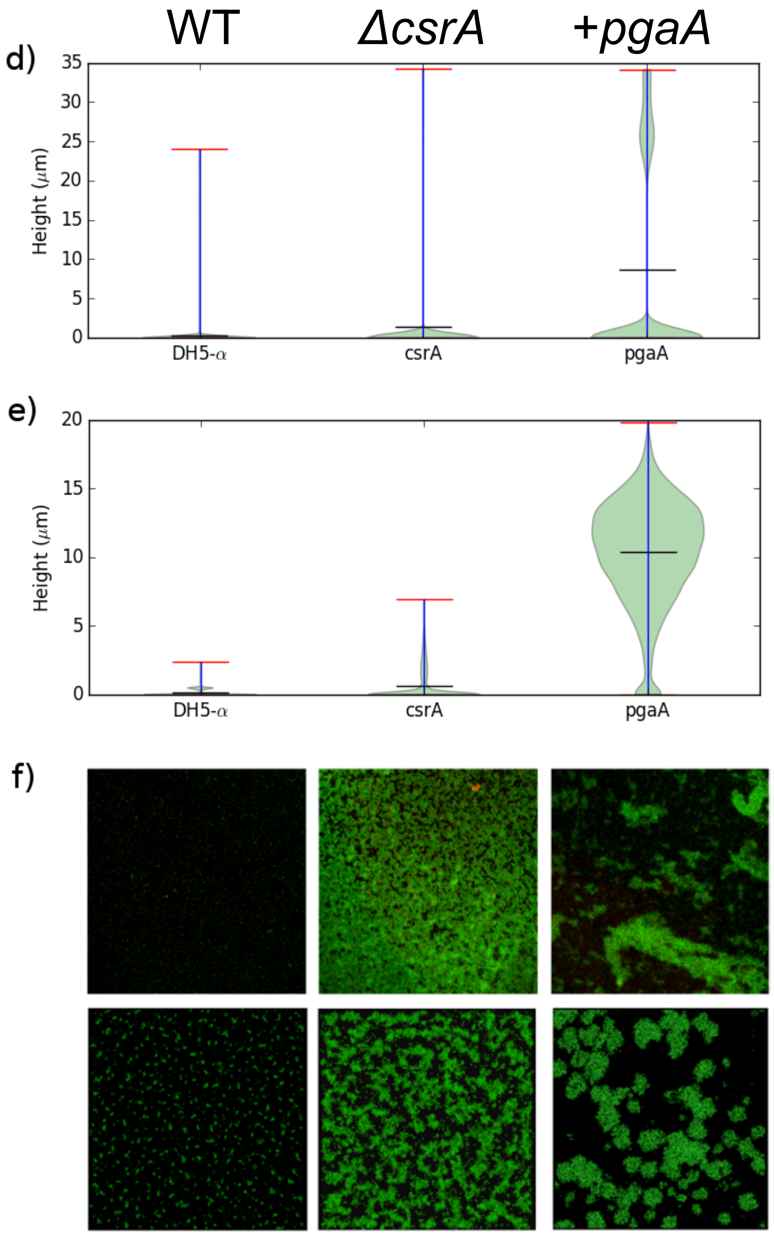


Asymmetrical cell-cell adhesion relative to cell-surface results in different biofilm architecture

Strong cell-surface + weak cell-cell = flat and uniform

Weak cell-surface + strong cell-cell = lumpy and irregular

Case Study: Biofilm formation



height profile

Experiment

Simulation

microscope
top view

Experiment

Simulation

ACS Synthetic Biology

Research Article

pubs.acs.org/synthbio

Simbiotics: A Multiscale Integrative Platform for 3D Modeling of Bacterial Populations

Jonathan Naylor,[†] Harold Fellermann,[†] Yuchun Ding,[†] Waleed K. Mohammed,[‡] Nicholas S. Jakubovics,[‡] Joy Mukherjee,[§] Catherine A. Biggs,[§] Phillip C. Wright,[∥] and Natalio Krasnogor[*,†]

[†]Interdisciplinary Computing and Complex Biosystems (ICOS) research group, School of Computing Science, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K.

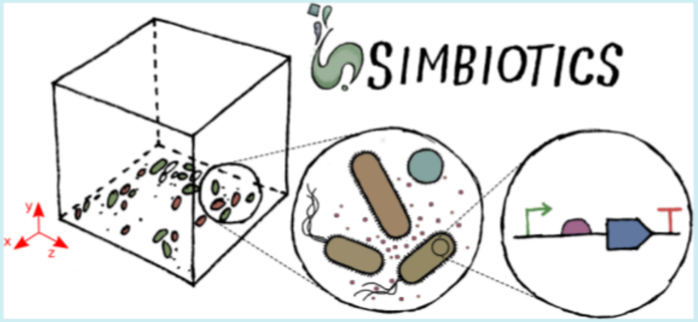
[‡]School of Dental Sciences, Newcastle University, Newcastle upon Tyne NE2 4BW, U.K.

[§]Department of Chemical and Biological Engineering, University of Sheffield, Sheffield S10 2TN, U.K.

[∥]School of Chemical Engineering and Advanced Materials, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K.

Supporting Information

ABSTRACT: Simbiotics is a spatially explicit multiscale modeling platform for the design, simulation and analysis of bacterial populations. Systems ranging from planktonic cells and colonies, to biofilm formation and development may be modeled. Representation of biological systems in Simbiotics is flexible, and user-defined processes may be in a variety of forms depending on desired model abstraction. Simbiotics provides a library of modules such as cell geometries, physical force dynamics, genetic circuits, metabolic pathways, chemical diffusion and cell interactions. Model defined processes are integrated and scheduled for parallel multithread and multi-CPU execution. A virtual lab provides the modeler with analysis modules and some simulated lab equipment, enabling automation of sample interaction and data collection. An extendable and modular framework allows for the platform to be updated as novel models of bacteria are developed, coupled with an intuitive user interface to allow for model definitions with minimal programming experience. Simbiotics can integrate existing standards such as SBML, and process microscopy images to initialize the 3D spatial configuration of bacteria consortia. Two case studies, used to illustrate the platform flexibility, focus on the physical properties of the biosystems modeled. These pilot case studies demonstrate Simbiotics versatility in modeling and analysis of natural systems and as a CAD tool for synthetic biology.

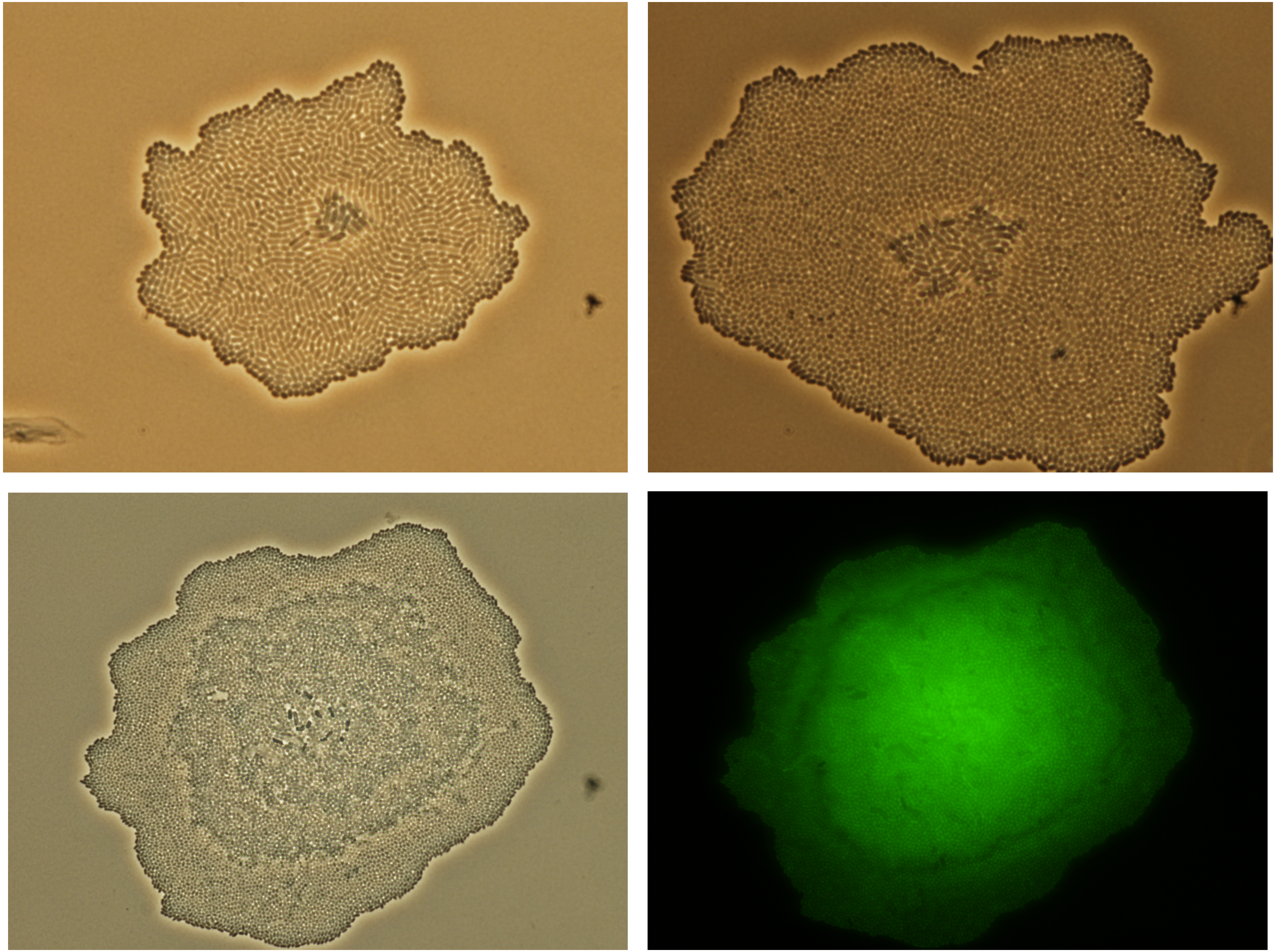


KEYWORDS: *bacterial population, simulation, multiscale, biofilm, agent-based model, interaction*

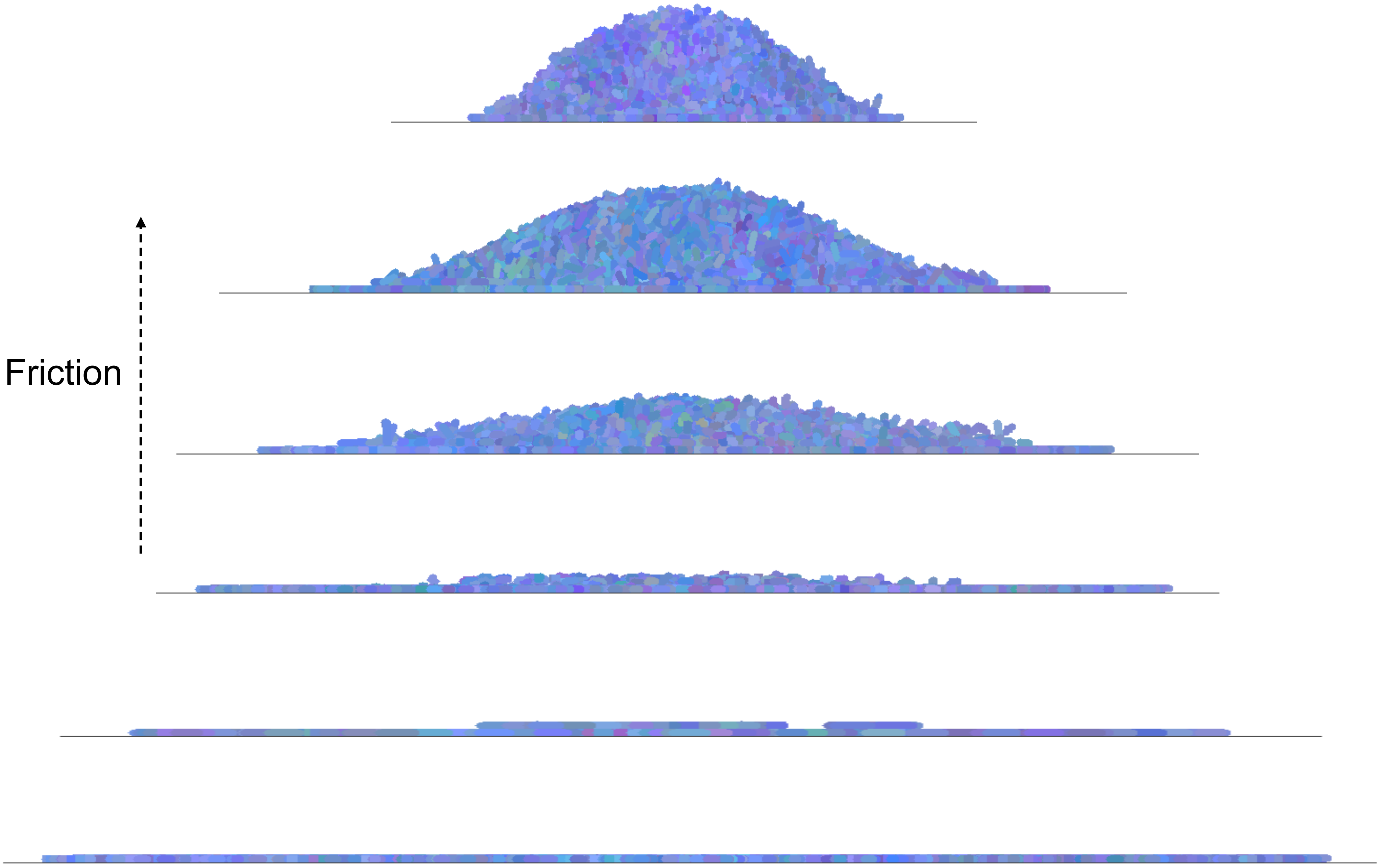
Bacterial colonies are networks of interacting cells that

design, synthesis and analysis of such synthetic systems is time

Colonies of rod-shaped cells (microscopy images)

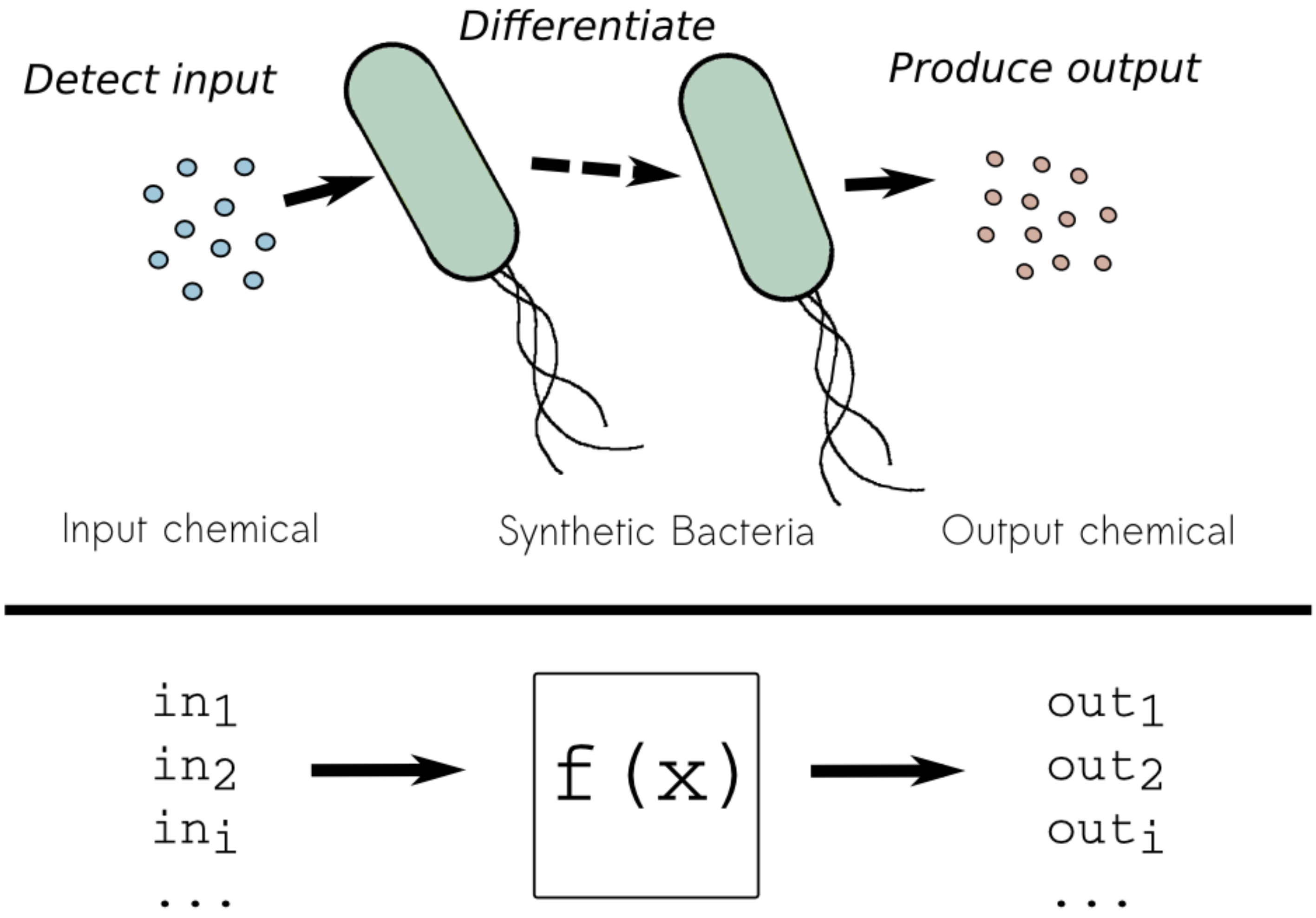


Simulation of rod colony



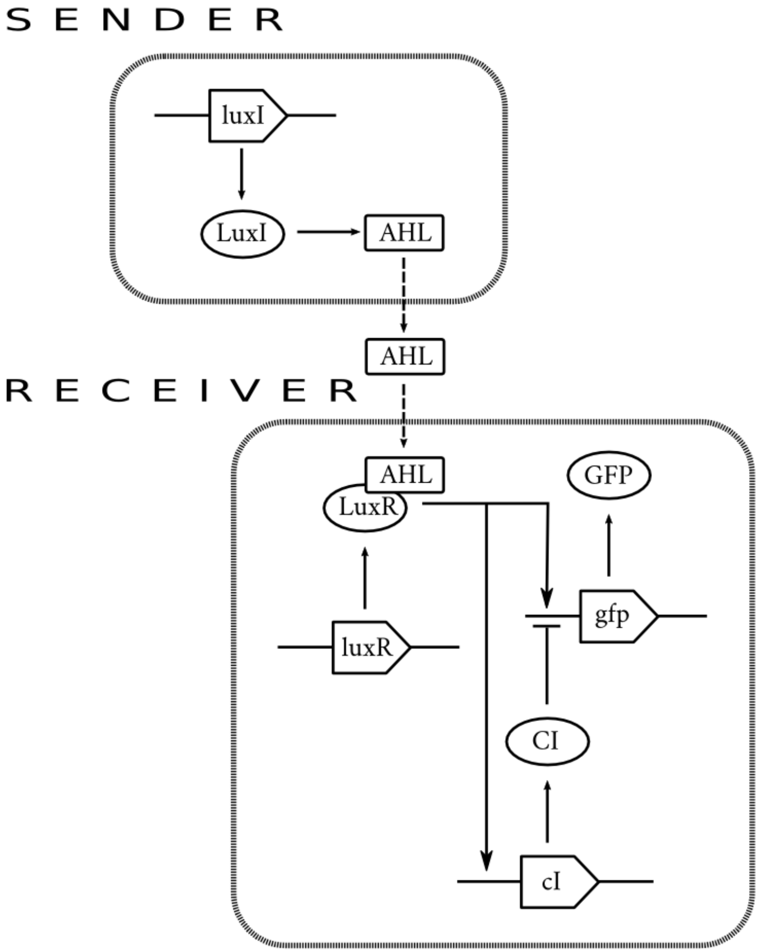
Pattern formation

In collaboration with:
Francisco Campero-Romero, Plant Development Unit, Institute for Plant Biochemistry and Photosynthesis, Consejo Superior de Investigaciones Científicas, Universidadde Sevilla, Seville, Spain



Pattern formation

In collaboration with:
Francisco Campero-Romero, Plant Development Unit, Institute for Plant Biochemistry and Photosynthesis, Consejo Superior de Investigaciones Científicas, Universidadde Sevilla, Seville, Spain



$$\frac{dAHL}{dt} = v_1 - (k_1 * AHL) \tag{1}$$

$$\frac{dAHL}{dt} = -k_1 * AHL \tag{2}$$

$$\frac{dLuxR}{dt} = v_2 - (k_2 * LuxR) \tag{3}$$

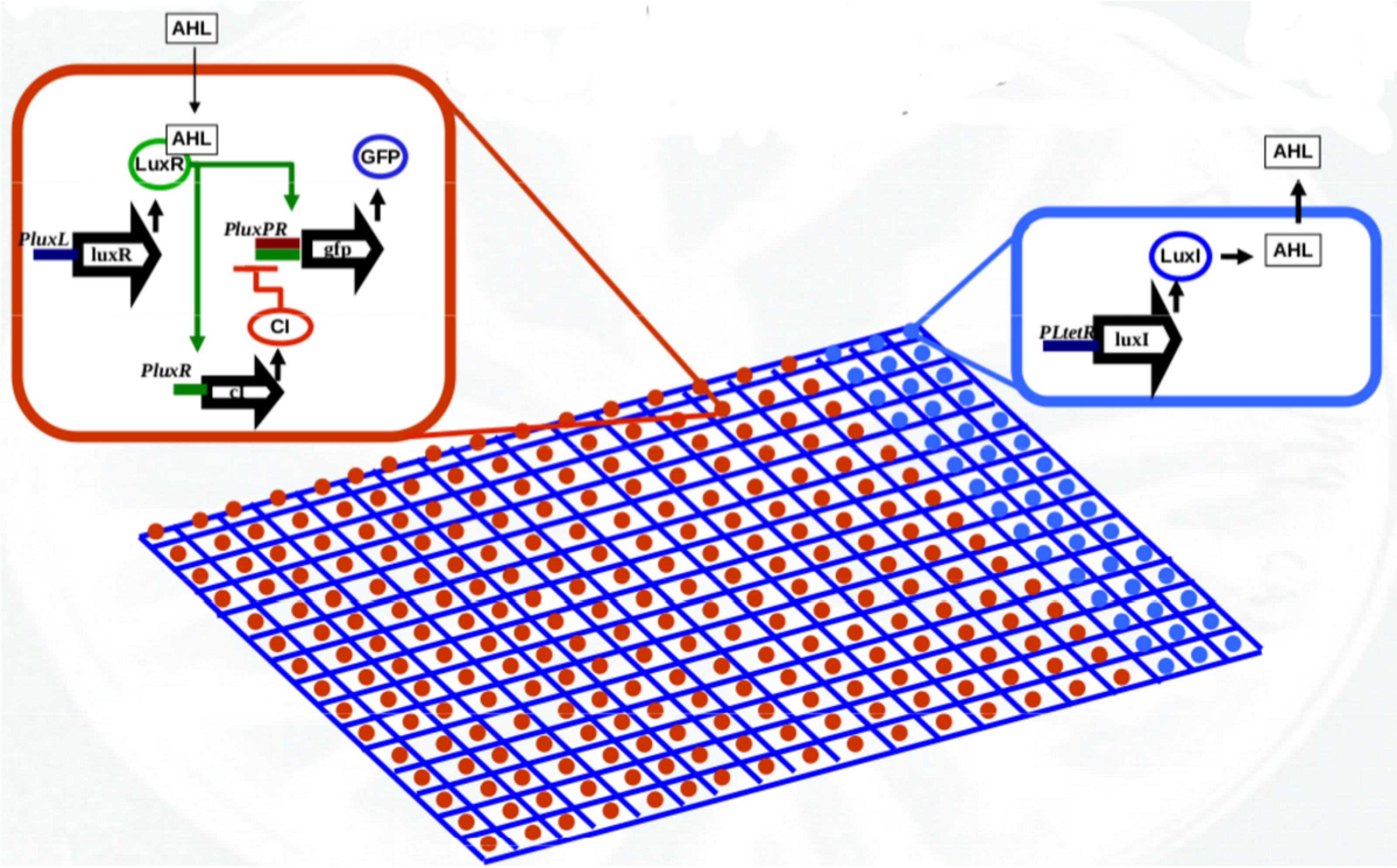
$$\frac{dLuxRAHL}{dt} = (k_3 * AHL * LuxR) - (k_4 * LuxRAHL) \tag{4}$$

$$\frac{dGFP}{dt} = \frac{v_3 * LuxRAHL}{LuxRAHL + k_5} * \frac{1}{1 + (CI/k_6)} - (k_6 * GFP) \tag{5}$$

$$\frac{dCI}{dt} = \frac{v_4 * LuxRAHL}{LuxRAHL + k_5} - k_7 * CI \tag{6}$$

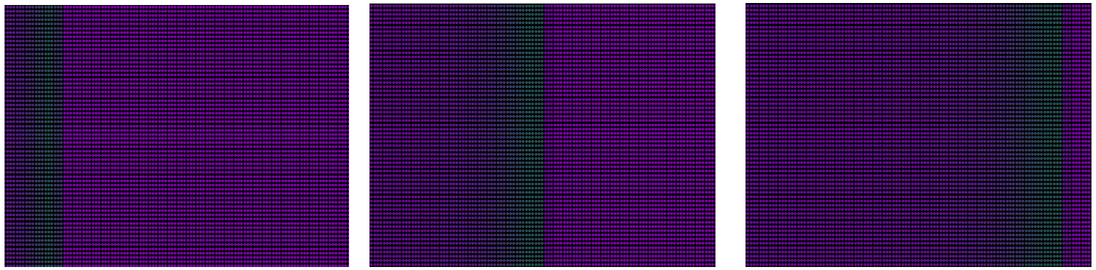
Pattern formation

In collaboration with:
Francisco Campero-Romero, Plant Development Unit, Institute for Plant Biochemistry and Photosynthesis, Consejo Superior de Investigaciones Científicas, Universidadde Sevilla, Seville, Spain



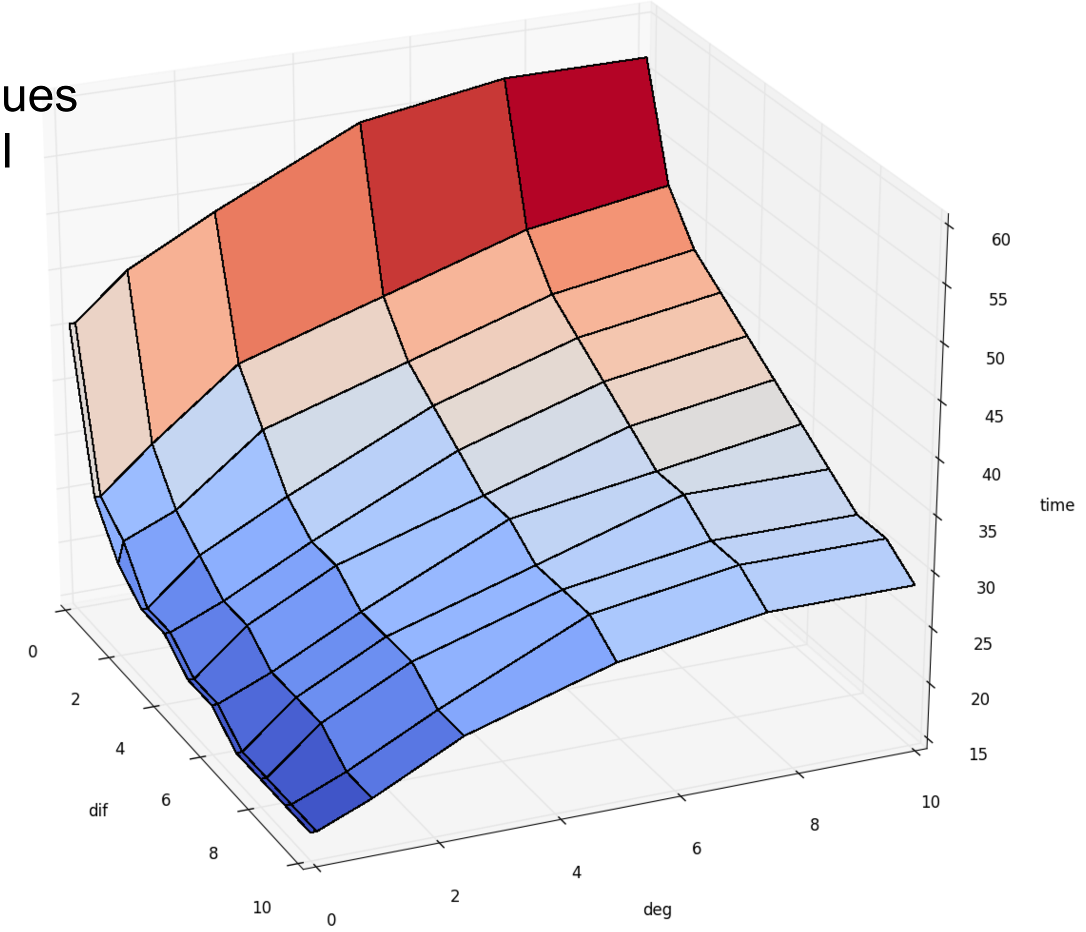
Pattern formation

Rendering of simulation showing GFP pulse propogate across the colony (cells are purple).



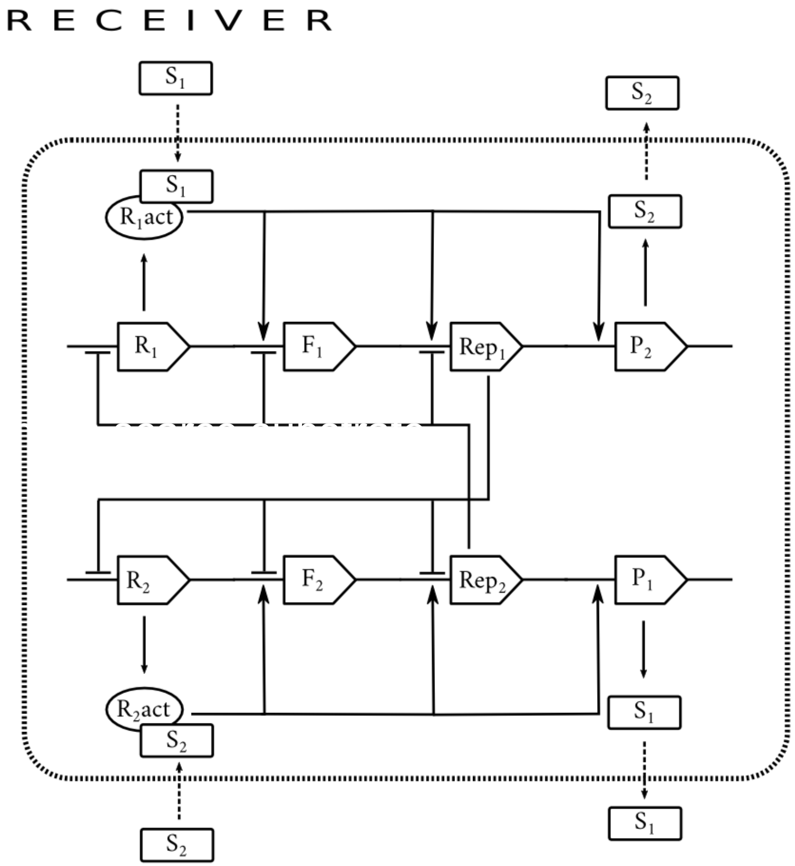
T = 5 T = 20 T = 40

Pulse velocity was measured for different values of the signal *diffusion coefficient* **D** and signal *degradation coefficient* **K**



Pattern formation

The second system of study is a Turing-like pattern formation system. There is only one species of cell, the receiever, which has parallel gene circuits either expressing flourescent protein F1 or F2, depending on which input singla it receives (S1 or S2). The circuit also responds by producing the other signal – this results in the formation of stripes of F1 and F2 expression.



$$\frac{dR_1}{dt} = \frac{v_1}{1+\frac{Rep_2}{k_i}} - (k_r * R_1) - (k_c * R_1 * S_1) \tag{7}$$

$$\frac{dR_2}{dt} = \frac{v_1}{1+\frac{Rep_1}{k_i}} - (k_r * R_1) - (k_c * R_2 * S_2) \tag{8}$$

$$\frac{dF_1}{dt} = v_2 * \frac{R_1act}{R_1act + km} * \frac{1}{1+\frac{Rep_2}{k_i}} - (k_d * F_1) \tag{9}$$

$$\frac{dF_2}{dt} = v_2 * \frac{R_2act}{R_2act + km} * \frac{1}{1+\frac{Rep_1}{k_i}} - (k_d * F_2) \tag{10}$$

$$\frac{dR_1act}{dt} = (k_c * R_1 * S_1) - (k_d * R_1act) \tag{11}$$

$$\frac{dR_2act}{dt} = (k_c * R_2 * S_2) - (k_d * R_2act) \tag{12}$$

$$\frac{dRep_1}{dt} = v_3 * \frac{R_1act}{R_1act + k_m} * \frac{1}{1+\frac{Rep_2}{k_i}} - (k_d * Rep_1) \tag{13}$$

$$\frac{dRep_2}{dt} = v_3 * \frac{R_2act}{R_2act + k_m} * \frac{1}{1+\frac{Rep_1}{k_i}} - (k_d * Rep_2) \tag{14}$$

$$\frac{dP_1}{dt} = v_4 * \frac{R_2act}{R_2act + k_m} - (k_d * P_1) \tag{15}$$

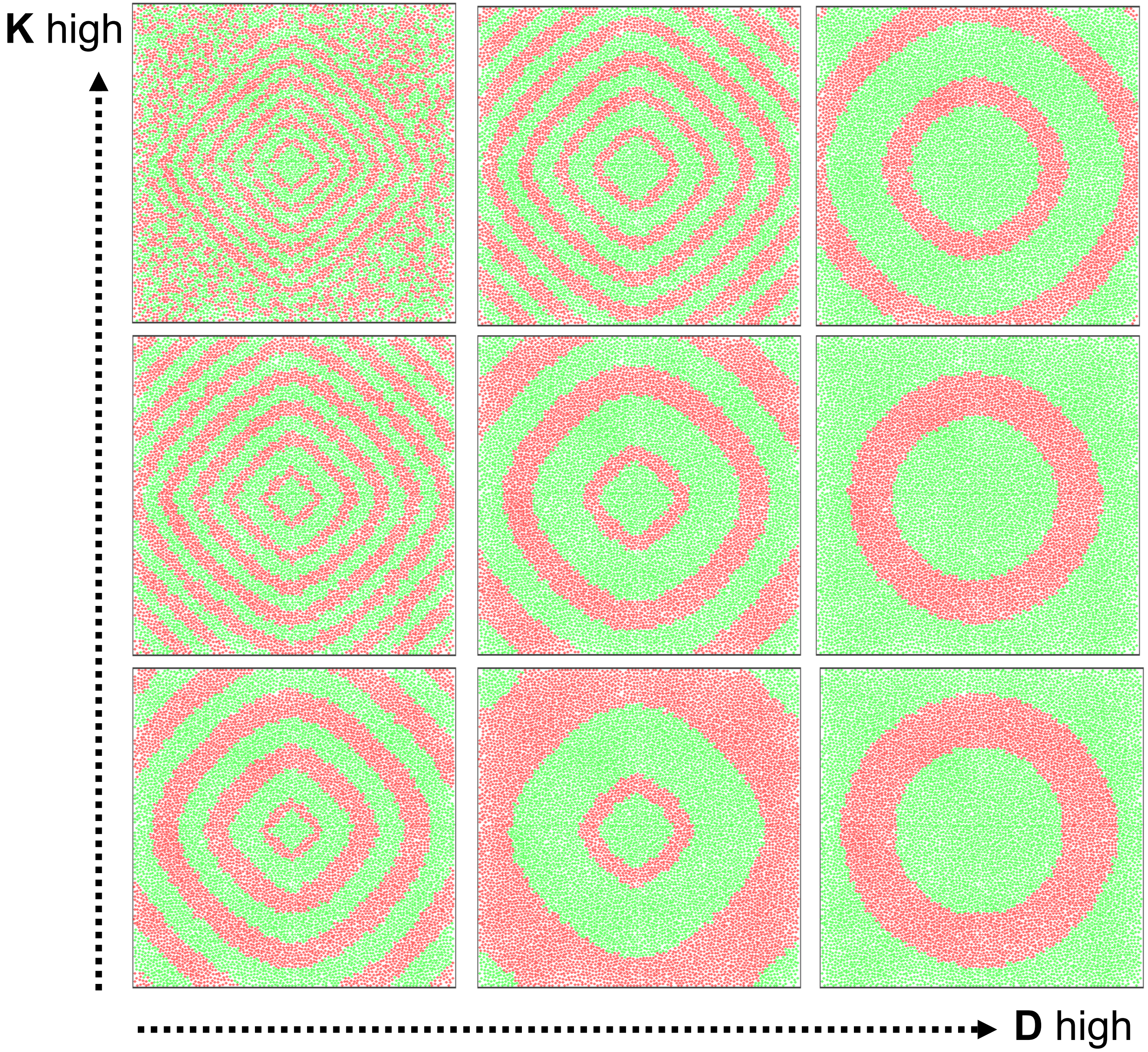
$$\frac{dP_2}{dt} = v_4 * \frac{R_1act}{R_1act + k_m} - (k_d * P_2) \tag{16}$$

$$\frac{dS_1}{dt} = (v_5 * P_1) - (k_s * P_1) - (k_c * R_1 * S_1) \tag{17}$$

$$\frac{dS_2}{dt} = (v_5 * P_2) - (k_s * P_2) - (k_c * R_2 * \bar{S}_2) \tag{18}$$

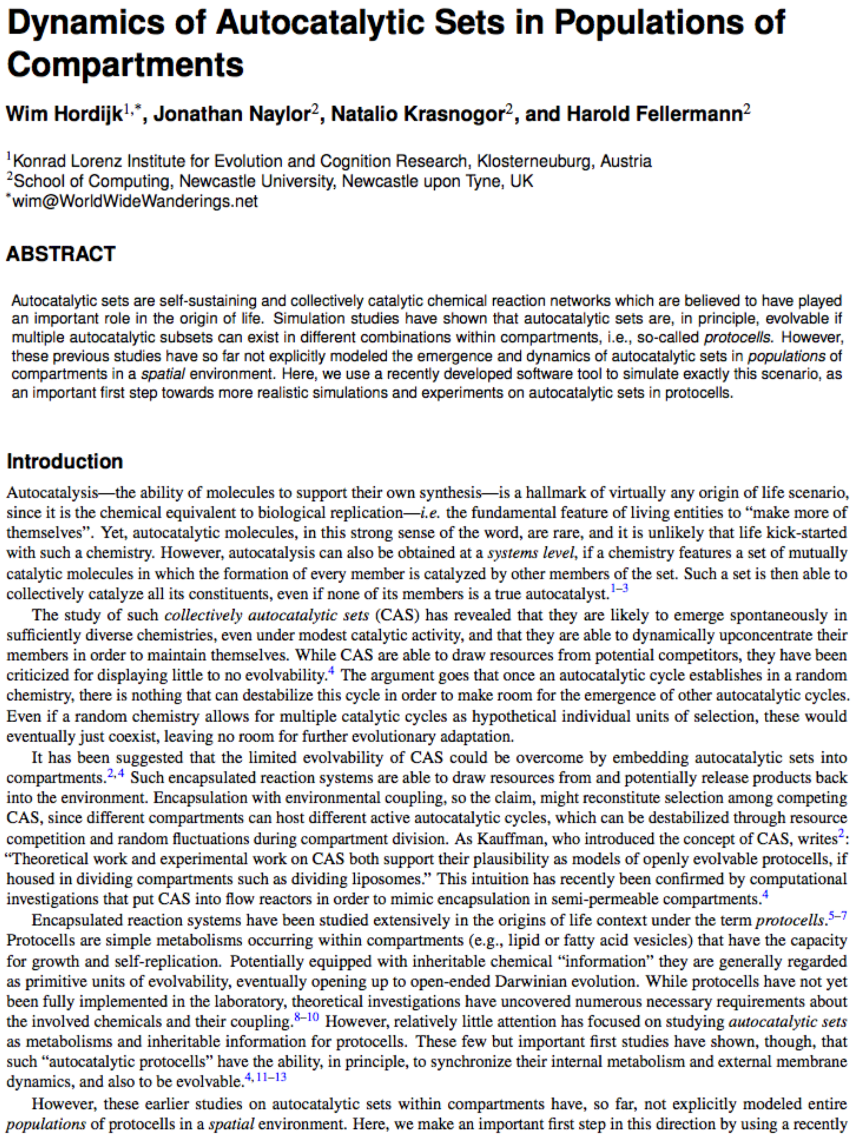
Pattern formation

Visualisations of colonys showing flourescent protein expression F1 (green) and F2 (red), for different signal *diffusion coefficient* **D** and signal *degradation coefficient* **K**. S1 and S2 are set to have equal **D** and **K** values. The system is induced by pipetting S1 at the center.



Autocatalytic sets

In collaboration with:
Wim Hordijk, Konrad Lorenz Institute for Evolution and Cognition Research, Klosterneuburg, Austria



Dynamics of Autocatalytic Sets in Populations of Compartments

Wim Hordijk[1,*], **Jonathan Naylor**[2], **Natalio Krasnogor**[2], **and Harold Fellermann**[2]

[1]Konrad Lorenz Institute for Evolution and Cognition Research, Klosterneuburg, Austria
[2]School of Computing, Newcastle University, Newcastle upon Tyne, UK
[*]wim@WorldWideWanderings.net

ABSTRACT

Autocatalytic sets are self-sustaining and collectively catalytic chemical reaction networks which are believed to have played an important role in the origin of life. Simulation studies have shown that autocatalytic sets are, in principle, evolvable if multiple autocatalytic subsets can exist in different combinations within compartments, i.e., so-called *protocells*. However, these previous studies have so far not explicitly modeled the emergence and dynamics of autocatalytic sets in *populations* of compartments in a *spatial* environment. Here, we use a recently developed software tool to simulate exactly this scenario, as an important first step towards more realistic simulations and experiments on autocatalytic sets in protocells.

Introduction

Autocatalysis—the ability of molecules to support their own synthesis—is a hallmark of virtually any origin of life scenario, since it is the chemical equivalent to biological replication—*i.e.* the fundamental feature of living entities to "make more of themselves". Yet, autocatalytic molecules, in this strong sense of the word, are rare, and it is unlikely that life kick-started with such a chemistry. However, autocatalysis can also be obtained at a *systems level*, if a chemistry features a set of mutually catalytic molecules in which the formation of every member is catalyzed by other members of the set. Such a set is then able to collectively catalyze all its constituents, even if none of its members is a true autocatalyst.[1–3]

The study of such *collectively autocatalytic sets* (CAS) has revealed that they are likely to emerge spontaneously in sufficiently diverse chemistries, even under modest catalytic activity, and that they are able to dynamically upconcentrate their members in order to maintain themselves. While CAS are able to draw resources from potential competitors, they have been criticized for displaying little to no evolvability.[4] The argument goes that once an autocatalytic cycle establishes in a random chemistry, there is nothing that can destabilize this cycle in order to make room for the emergence of other autocatalytic cycles. Even if a random chemistry allows for multiple catalytic cycles as hypothetical individual units of selection, these would eventually just coexist, leaving no room for further evolutionary adaptation.

It has been suggested that the limited evolvability of CAS could be overcome by embedding autocatalytic sets into compartments.[2,4] Such encapsulated reaction systems are able to draw resources from and potentially release products back into the environment. Encapsulation with environmental coupling, so the claim, might reconstitute selection among competing CAS, since different compartments can host different active autocatalytic cycles, which can be destabilized through resource competition and random fluctuations during compartment division. As Kauffman, who introduced the concept of CAS, writes[2]: "Theoretical work and experimental work on CAS both support their plausibility as models of openly evolvable protocells, if housed in dividing compartments such as dividing liposomes." This intuition has recently been confirmed by computational investigations that put CAS into flow reactors in order to mimic encapsulation in semi-permeable compartments.[4]

Encapsulated reaction systems have been studied extensively in the origins of life context under the term *protocells*.[5–7] Protocells are simple metabolisms occurring within compartments (e.g., lipid or fatty acid vesicles) that have the capacity for growth and self-replication. Potentially equipped with inheritable chemical "information" they are generally regarded as primitive units of evolvability, eventually opening up to open-ended Darwinian evolution. While protocells have not yet been fully implemented in the laboratory, theoretical investigations have uncovered numerous necessary requirements about the involved chemicals and their coupling.[8–10] However, relatively little attention has focused on studying *autocatalytic sets* as metabolisms and inheritable information for protocells. These few but important first studies have shown, though, that such "autocatalytic protocells" have the ability, in principle, to synchronize their internal metabolism and external membrane dynamics, and also to be evolvable.[4,11–13]

However, these earlier studies on autocatalytic sets within compartments have, so far, not explicitly modeled entire *populations* of protocells in a *spatial* environment. Here, we make an important first step in this direction by using a recently

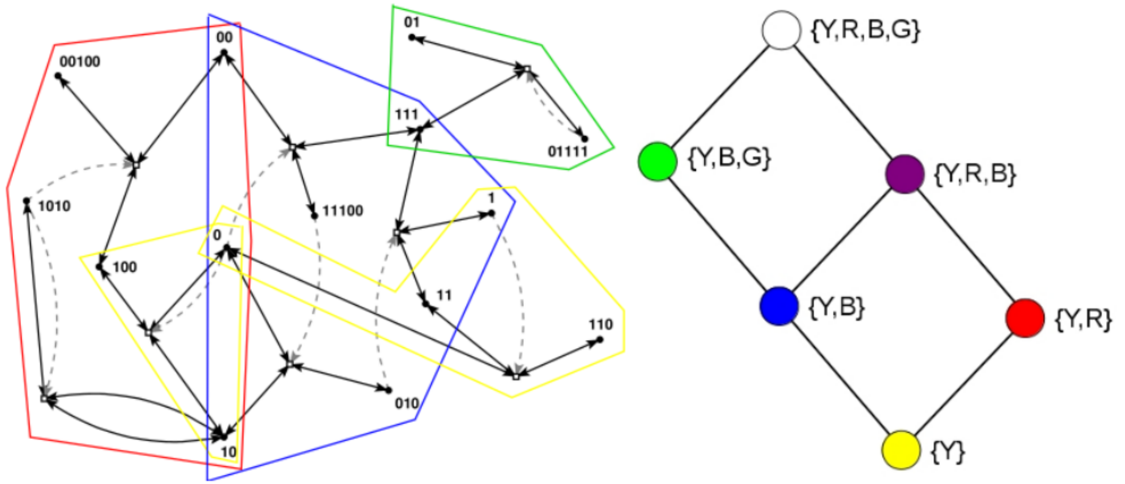


Figure 1. Left: An example of a RAF set as found in an instance of the binary polymer model. Black dots (labeled with bit strings) represent the molecule types, and white boxes represent reactions. Solid black arrows indicate molecules going into and coming out of a reaction, while dashed gray arrows indicate catalysis. Colored polygones indicate some of the RAF subsets. Right: The six closed RAFs (color coded) and their mutual subset relationships.

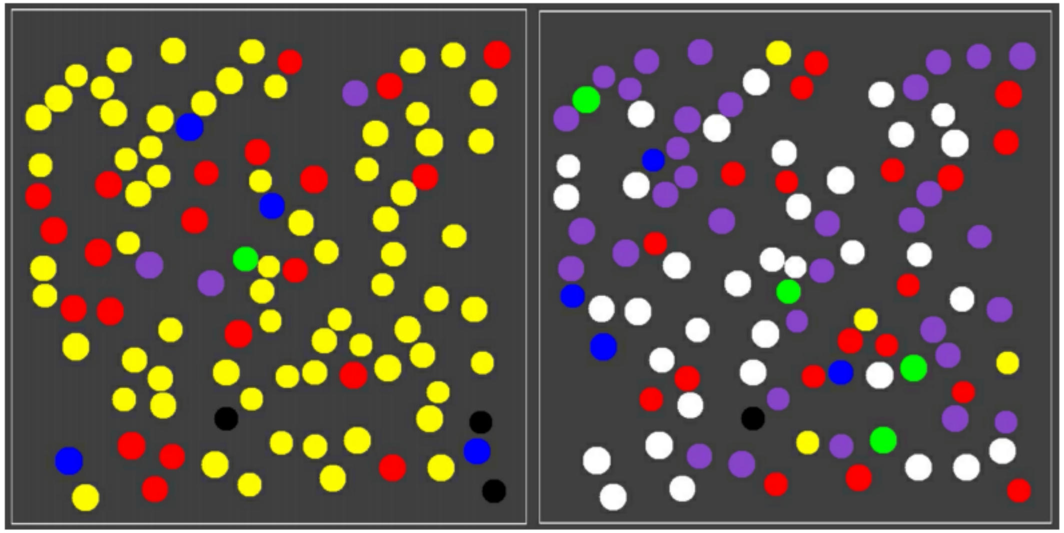
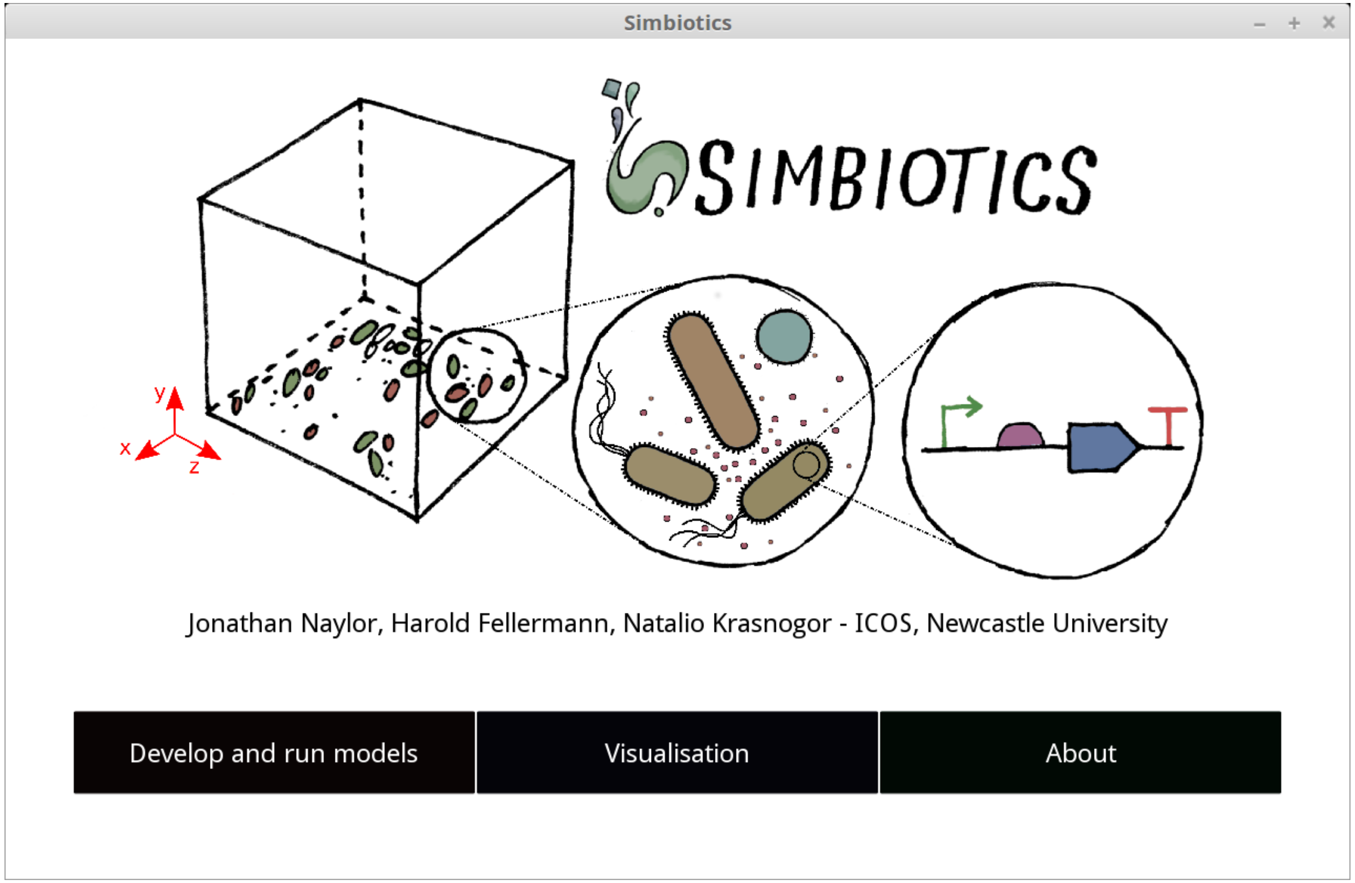
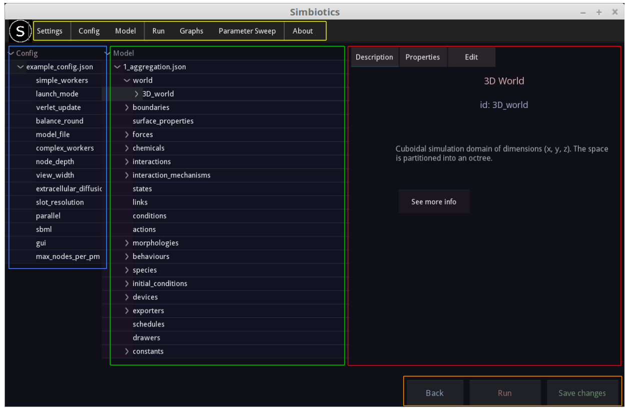


Figure 5. Two snapshots from a simulation with 100 compartments.

Easybiotics



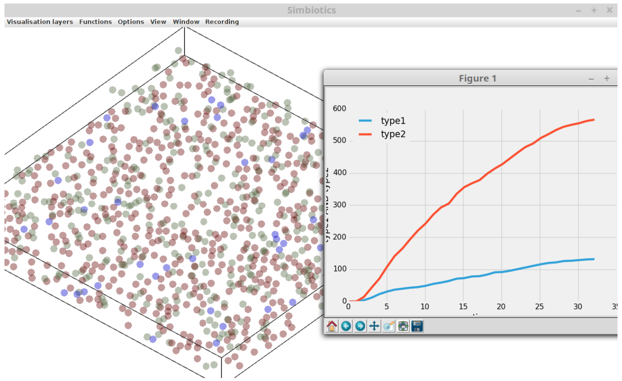
Jonathan Naylor, Harold Fellermann, Natalio Krasnogor - ICOS, Newcastle University



3D World

id: 3D_world

Cuboidal simulation domain of dimensions (x, y, z). The space is partitioned into an octree.

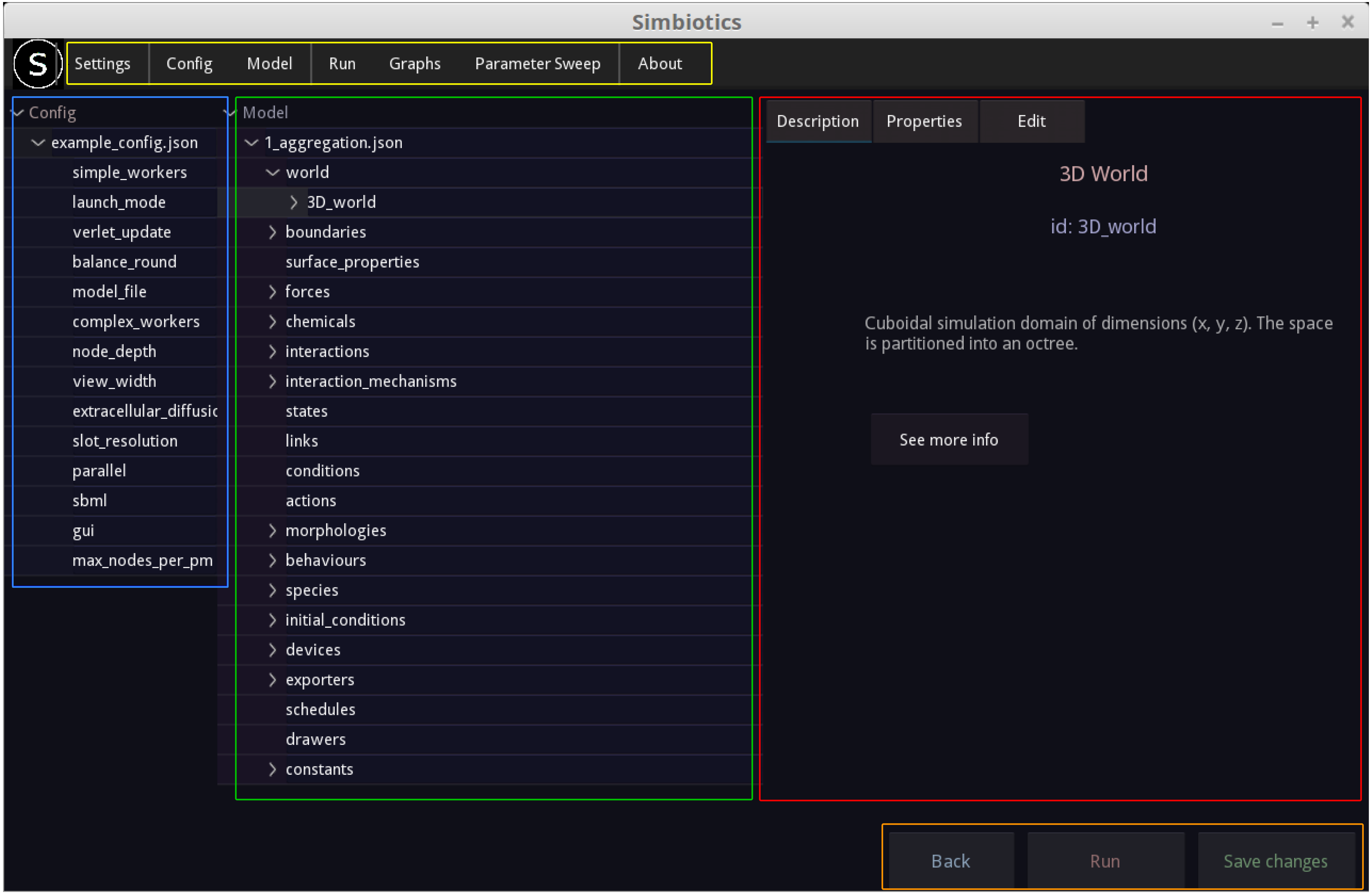


- Easybiotics allows for the development of Simbiotics models via an easy click-select graphical user interface – it does not require any programming experience to use

- The user can run simulations with/without live rendering and graph plotting

- Post-simulation visualisations can also be rendered in Easybiotics

Easybiotics – Model development environment



Config
- example_config.json
  - simple_workers
  - launch_mode
  - verlet_update
  - balance_round
  - model_file
  - complex_workers
  - node_depth
  - view_width
  - extracellular_diffusic
  - slot_resolution
  - parallel
  - sbml
  - gui
  - max_nodes_per_pm

Model
- 1_aggregation.json
  - world
    - 3D_world
  - boundaries
  - surface_properties
  - forces
  - chemicals
  - interactions
  - interaction_mechanisms
  - states
  - links
  - conditions
  - actions
  - morphologies
  - behaviours
  - species
  - initial_conditions
  - devices
  - exporters
  - schedules
  - drawers
  - constants

3D World

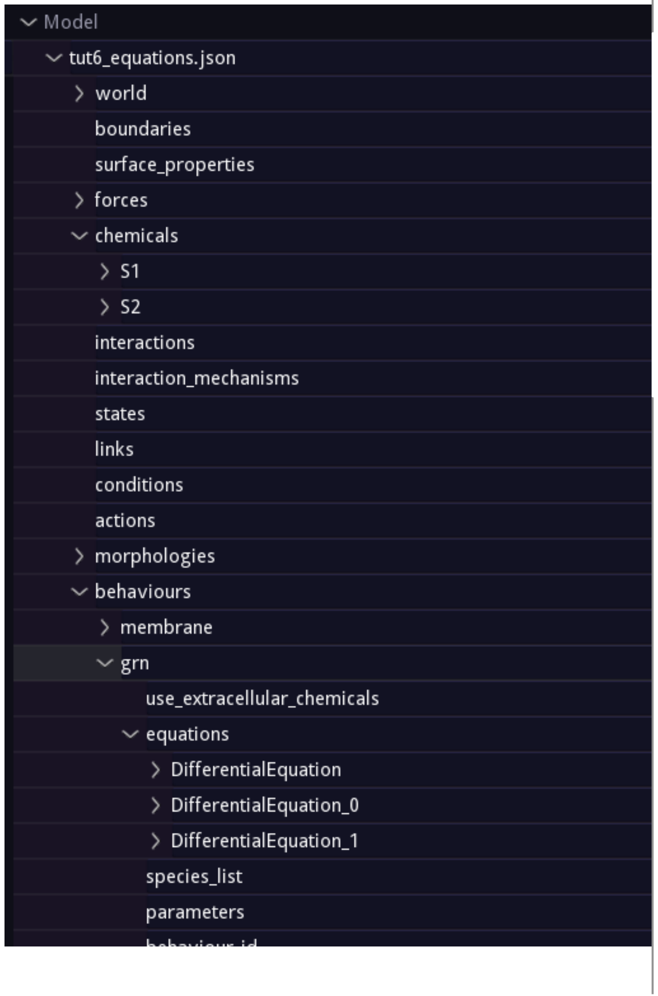
id: 3D_world

Cuboidal simulation domain of dimensions (x, y, z). The space is partitioned into an octree.

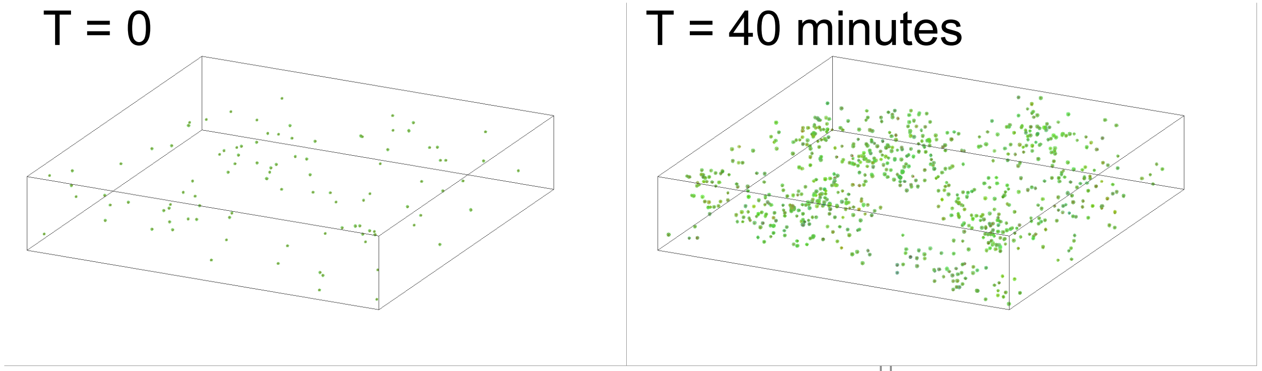
yellow = file bar   blue = config editor   green = model editor   red = display panel   orange = button panel

Easybiotics – Modelling example

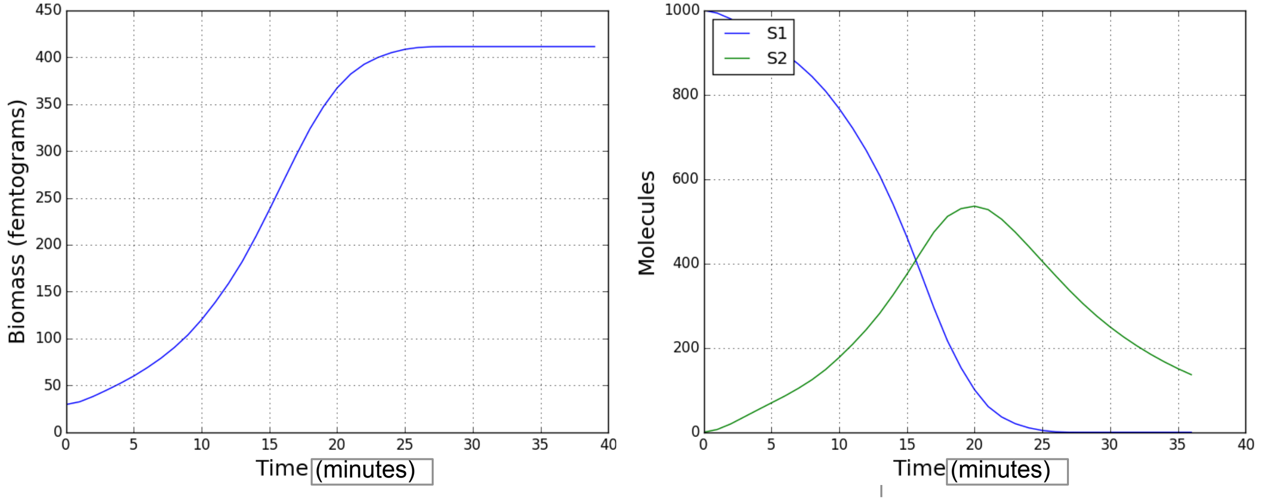
Model specification



```
⌄ Model
  ⌄ tut6_equations.json
    › world
      boundaries
      surface_properties
    › forces
    ⌄ chemicals
      › S1
      › S2
      interactions
      interaction_mechanisms
      states
      links
      conditions
      actions
    › morphologies
    ⌄ behaviours
      › membrane
      ⌄ grn
          use_extracellular_chemicals
        ⌄ equations
          › DifferentialEquation
          › DifferentialEquation_0
          › DifferentialEquation_1
          species_list
          parameters
          behaviour_id
```



Graphs showing biomass and chemical quantities over time



Example model where a cellular species grows due to a limited environmental nutrient S1. Growing cells produce and secrete S2, which degrades quite fast in the extracellular space. When S1 is depleted both growth and S2 production stop.

The intracellular dynamics are specified as a set of ordinary different equations.

Get it and Play!

Simbiotics

Simbiotics and Easybiotics:
Both tools are part of the same download, click on download repository at the link below:
https://bitbucket.org/simbiotics/simbiotics/downloads/

Alternatively you can clone the mercurial repository with:

```
hg clone https://simbiotics@bitbucket.org/simbiotics/simbiotics
```

The download includes the **user_manual.pdf** file, which contains full details on installation, tool use and software architecture. Details on implementations of tool functionality can be found on our ACS paper:
https://pubs.acs.org/doi/abs/10.1021/acssynbio.6b00315

We will now run through the installation process for Simbiotics and Easybiotics. There are then a series of tutorial exercises in the user_manual which you can work through to get a hang of the tool!

Get it and Play!

Simbiotics

Simbiotics

If you don't want to use the SBML integration feature, or the Easybiotics GUI, then there are no dependencies. (just do steps 1 and 5)

Simbiotics is developed in Java 1.7

SBML Integration Dependencies:

LibSBML	http://sbml.org/Software/libSBML
LibSBMLsim	http://fun.bio.keio.ac.jp/software/libsbmlsim/

1) download the Simbiotics archived file and extract it
2) install LibSBML and LibSBMLsim on your computer
3) locate libsbmlj.so and libsbmlsimj.so
4) copy and paste them into simbiotics/jars
5) open a terminal and cd into the main simbiotics directory, and enter the command: `java -jar simbiotics.jar`

Get it and Play!

Easybiotics

Easybiotics:

Dependencies:

Python >=2.7	https://www.python.org/downloads/
python-kivy	https://kivy.org/#home
python-matplotlib	https://matplotlib.org/
python-pandas	https://pandas.pydata.org/

1) once you have python, you can use *apt* or *pip* to install, eg:

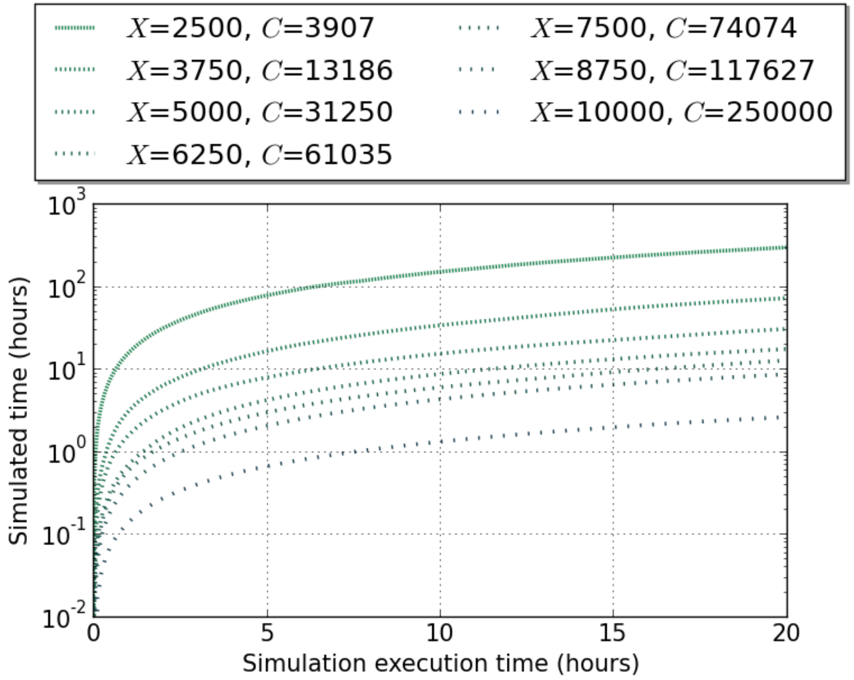
```
sudo apt-get install python-matplotlib python-pandas python-kivy
```

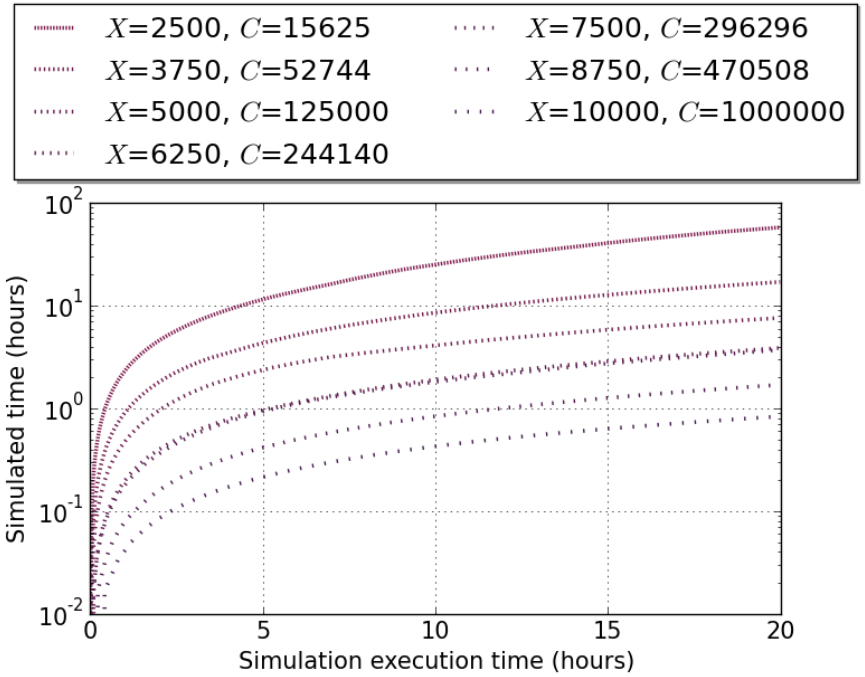
2) when all dependencies are installed, open a terminal and cd into the main simbiotics directory, and enter the command:

```
./run_simbiotics
```

...and you are good to go!

Performance



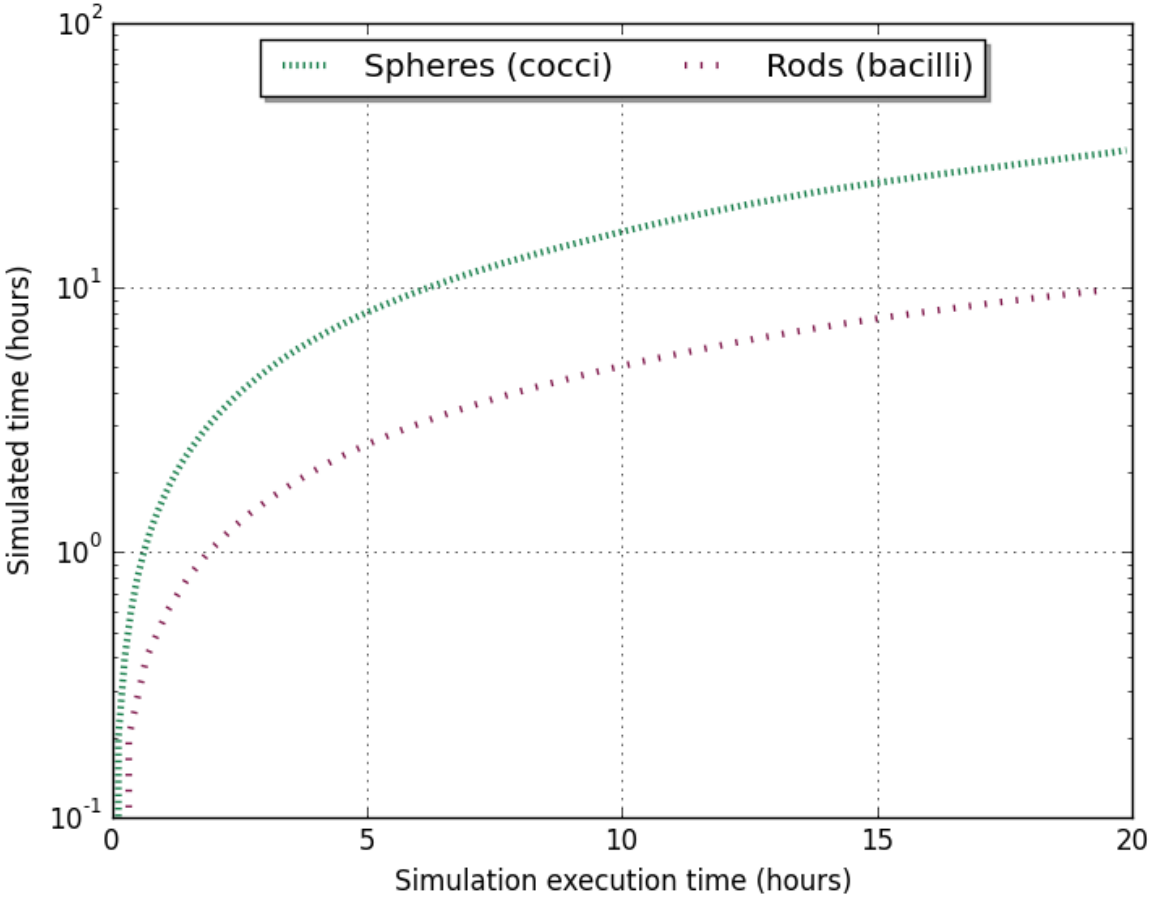


X = side length of cubic domain (μm)
[X = 10000 is equal to 1mL]

C = number of cells (spherical cells)

Performance

Simulating cells as spheres vs rods



Thanks for listening!

Jonathan Naylor
j.r.d.naylor@ncl.ac.uk

Harold Fellermann, Yuchun Ding, Natalio Krasnogor*
Interdisciplinary Computing and Complex BioSystems (ICOS)
School of Computing Science
Newcastle University

Dental school collaboration
School of Dental Sciences, Newcastle University
Waleed Mohammed,
Nick Jakubovics
Synthetic E.coli collaboration
Department of Chemical Engineering, University of Sheffield
Joy Mukherjee, Catherine Biggs
Philip Wright (School of Agriculture and Engineering, Newcastle University)

EPSRC
Engineering and Physical Sciences Research Council

ROADBLOCK

Example 1 (Easybiotics Tutorial 4 in User Manual)

Dominance of different interaction types

- In this model, we have a cell species which has two types of interaction with the substratum (bottom domain boundary). We create 1000 cells in a well mixed distribution.

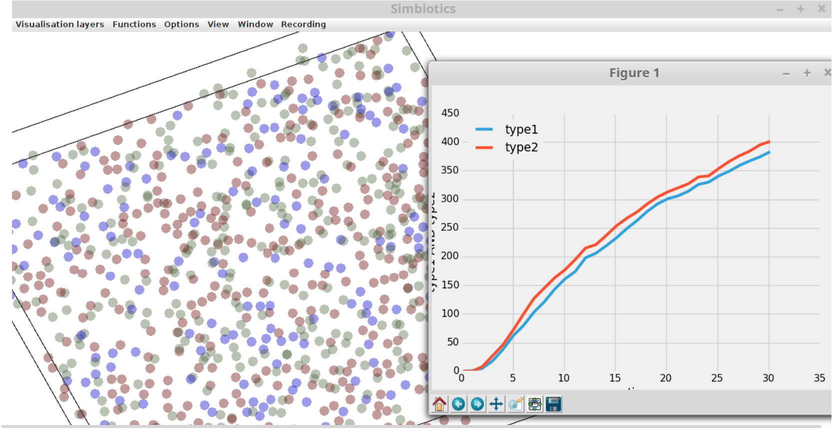
- The two interactions, *type1* and *type2*, which occur at different rates, ***r1*** and ***r2***.

- We set the maximum number of interactions a cell can have with the substatum to be 3

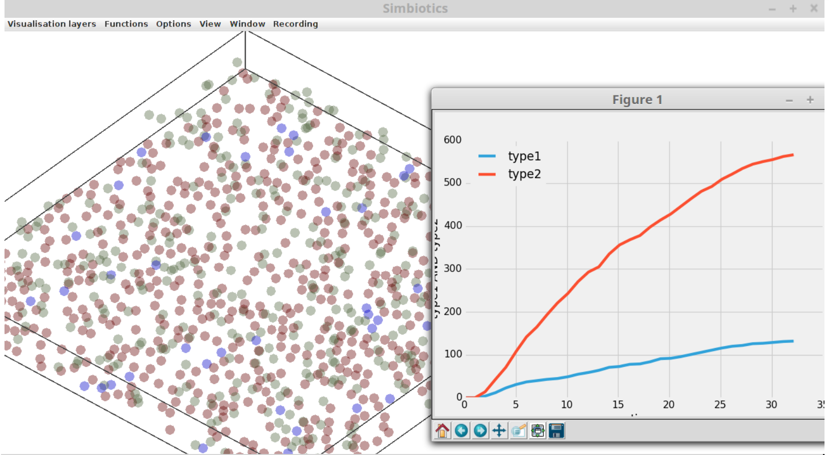
- We colour the cell blue if it has more than 1 interaction of *type1*, and red if it has more than 1 of interaction *type2*

- We show the resulting populations for different values of ***r1*** and ***r2***, the graphs show the number of cells which are in the blue/red state over time

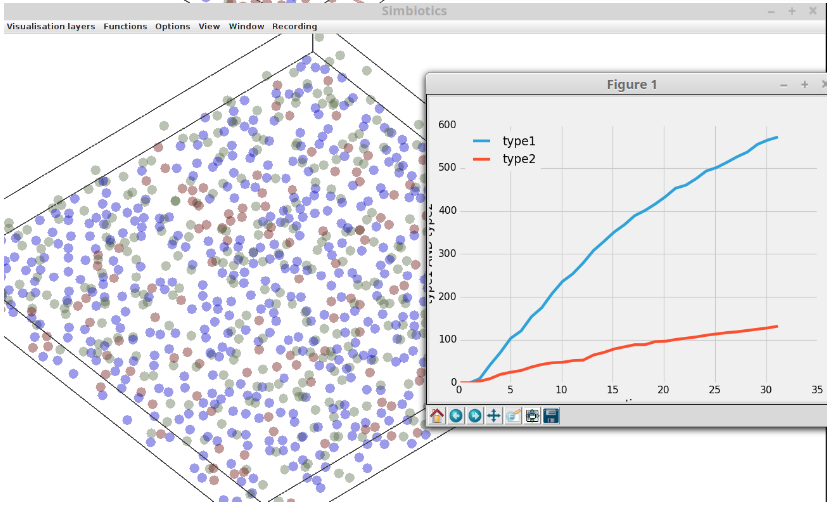
$r1 = 50$
$r2 = 50$



$r1 = 25$
$r2 = 75$


Simbiotics
Visualisation layers Functions Options View Window Recording
Figure 1
type1
type2
600
500
400
300
200
100
0
0
5
10
15
20
25
30

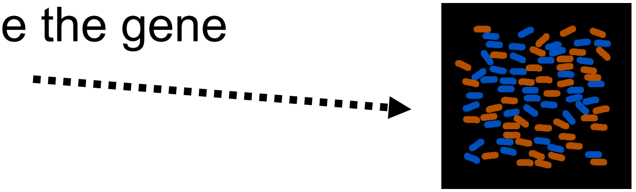
$r1 = 75$
$r2 = 25$



Example 2 (Easybiotics Tutorial 7 in User Manual)

Conjugation in a growing population

- In this model we have a single cell species, and create 80 of them in the center of the domain. We set 40 of them to have *gene A*, and 40 of them not to have the gene



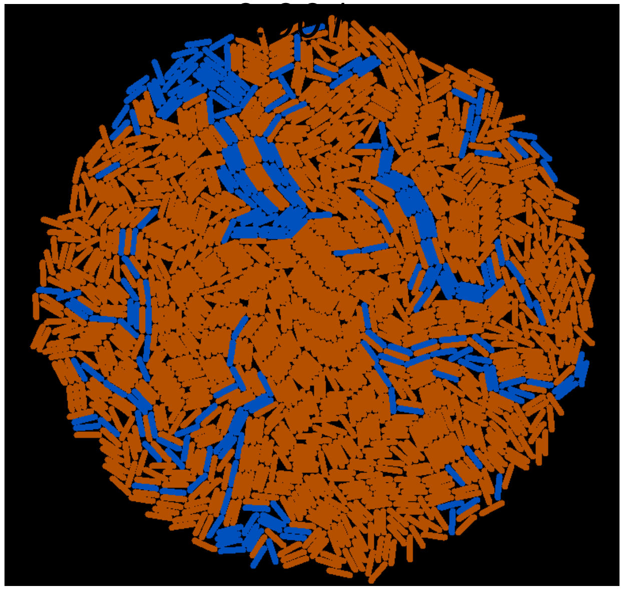
- We colour the cell red if it has *gene A,* and blue if it does not

- Cells with *gene A* can conjugate with the cells which they are in physical contact with, conjugation occurs at rate ***r***

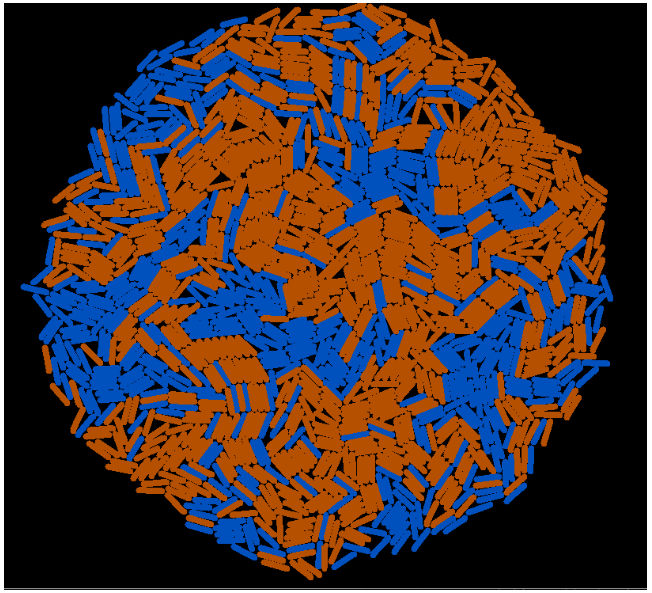
- The colony grows assuming a constant nutient source

- We show the resulting populations for different values of ***r***

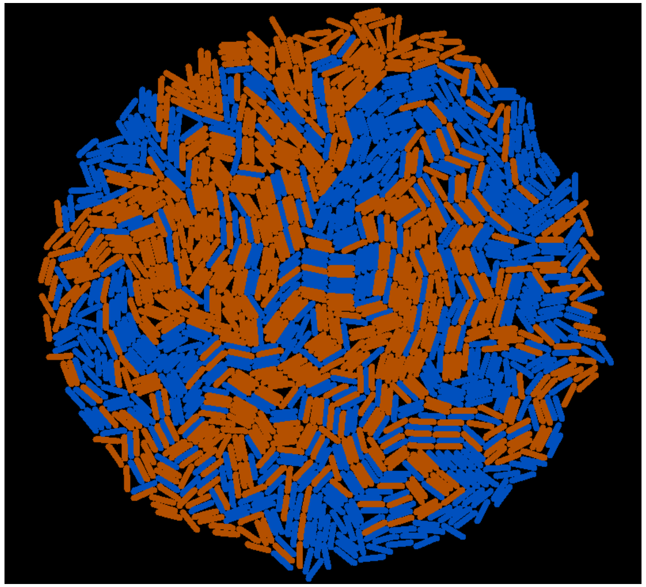
r = *0.1*



r = *0.01*



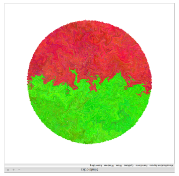
r = *0.001*



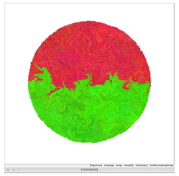
Reproducing literature results

CellModeller4 – Fractal rod-shaped cell colony boundaries

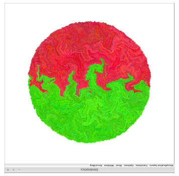
(https://pubs.acs.org/doi/pdf/10.1021/sb400030p)



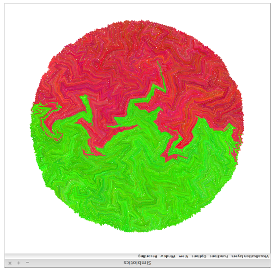
L *= 1.0μm*



L *= 1.5μm*



L *= 2.0μm*



L *= 2.5μm*

Cell length (**L**)

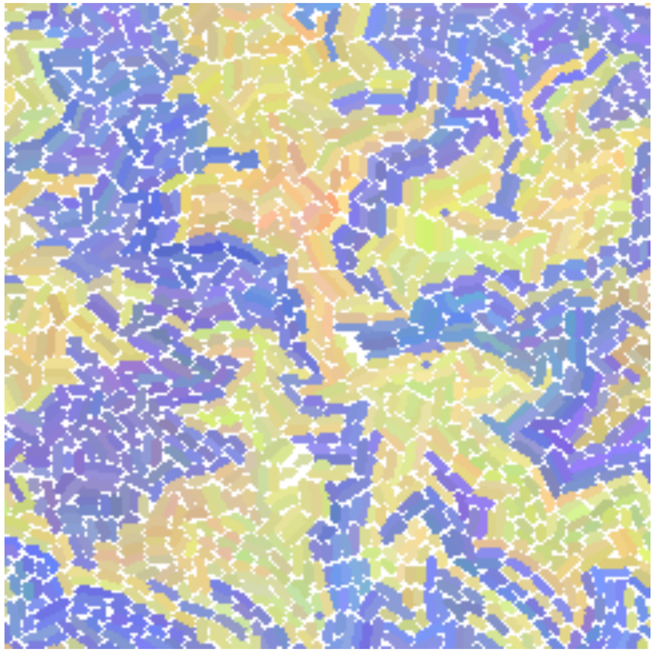
λThe system is seeded by 1 green cell and 1 red cell (both rod-shaped) which are parallel to each other, place in the center of the domain 20 micrometers apart.

λCells grow at a constant rate assuming a constant nutrient source

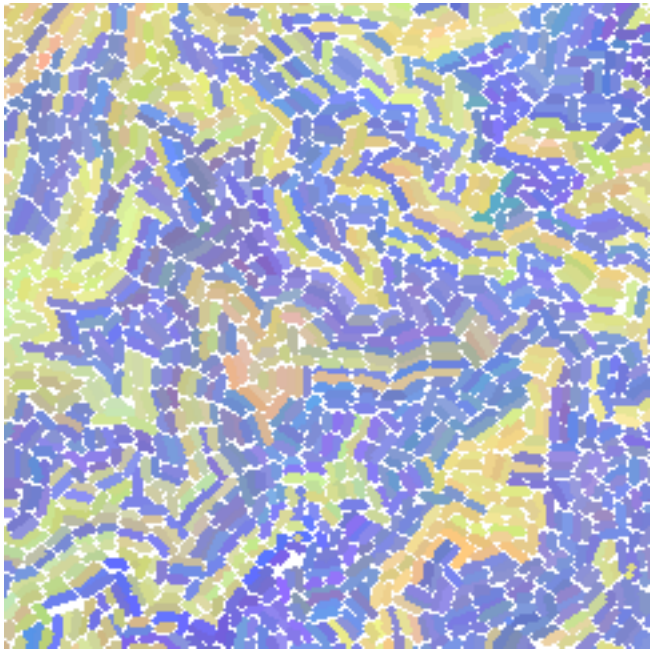
λAs cell length increases, as does the fractal dimension

Reproducing literature results

gro – Ecological growth models (https://pubs.acs.org/doi/pdf/10.1021/acssynbio.7b00003)



Mutualism



Cooperation

Competition

Mutalistism: species only interact through physical shoving
Cooperation: species rely on food produced by the other
Competition: species produce a toxin which prevents the other species from growing

λAll systems are initialised with a random distribution of 10 yellow cells and 10 purple cells

λWhen cooperating, an abundance of food tends to a mutualistic population distribution. When competing, an abundance of toxin leads to separate balanced populations or one of the species becoming extinct

Some other (arbitrary) models

The following slides contain arbitrary models developed in Simbiotics, some are prototypes for case studies, others are tests of functionality and exploration of tool capacity

For more examples of the tool, please see the user manual (see slide 33) where you can find a series of tutorials

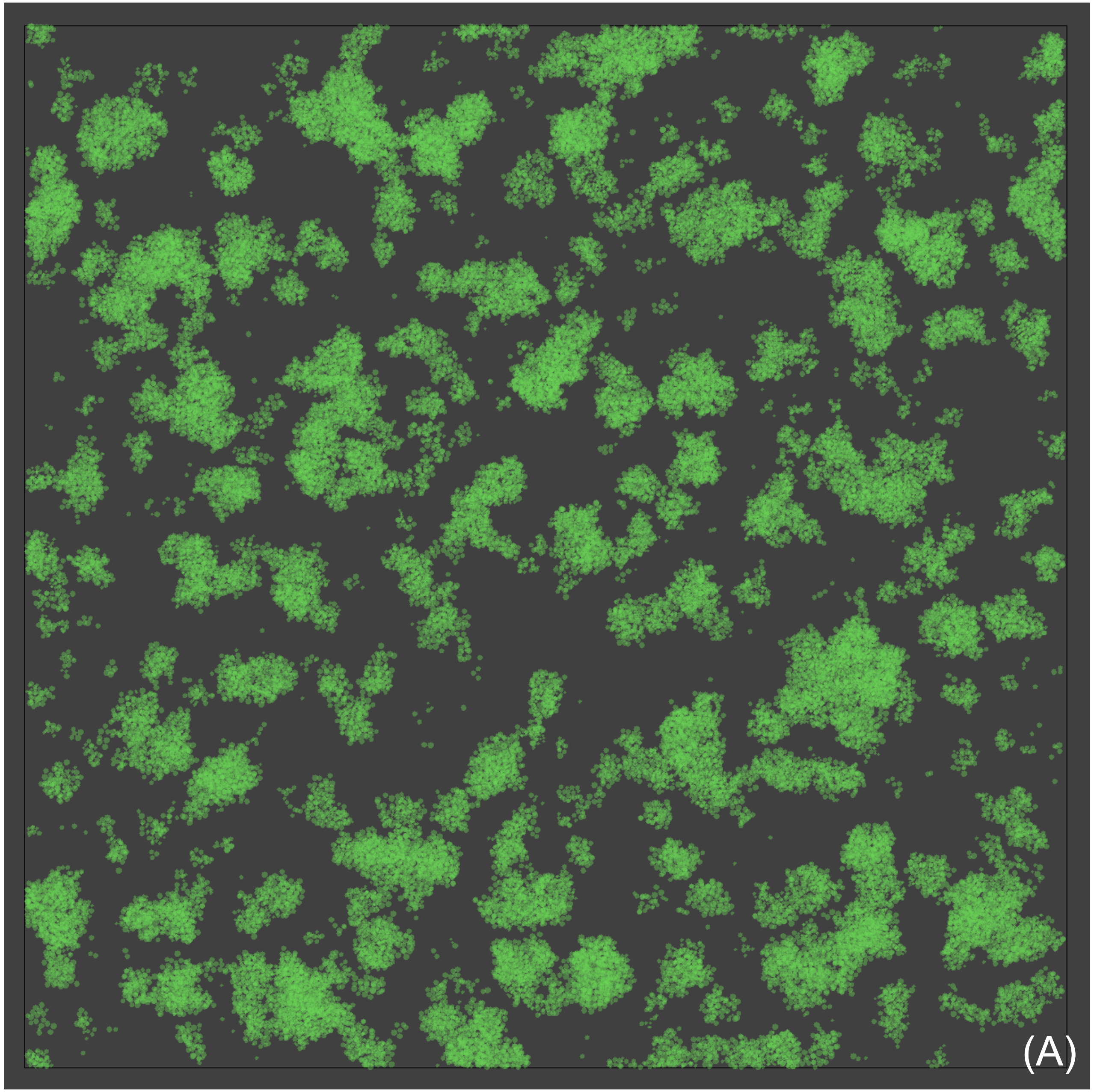
3D single species biofilm model (from Biofilm case study)

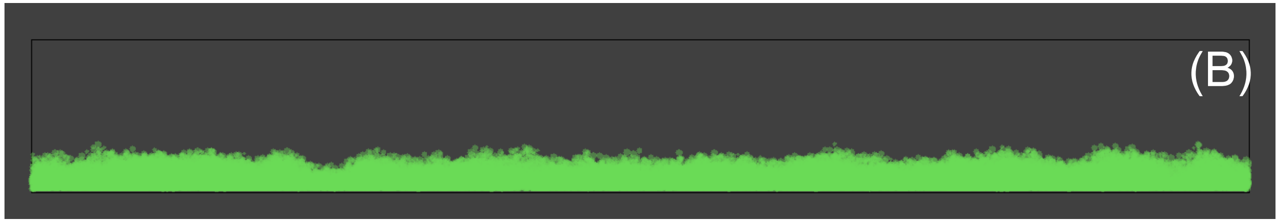
There's an influx of planktonic cells from the top of the domain

These planktonic cells may adhere to the surface (only these cells are visualised)

A constant nutrient concentration is assumed.

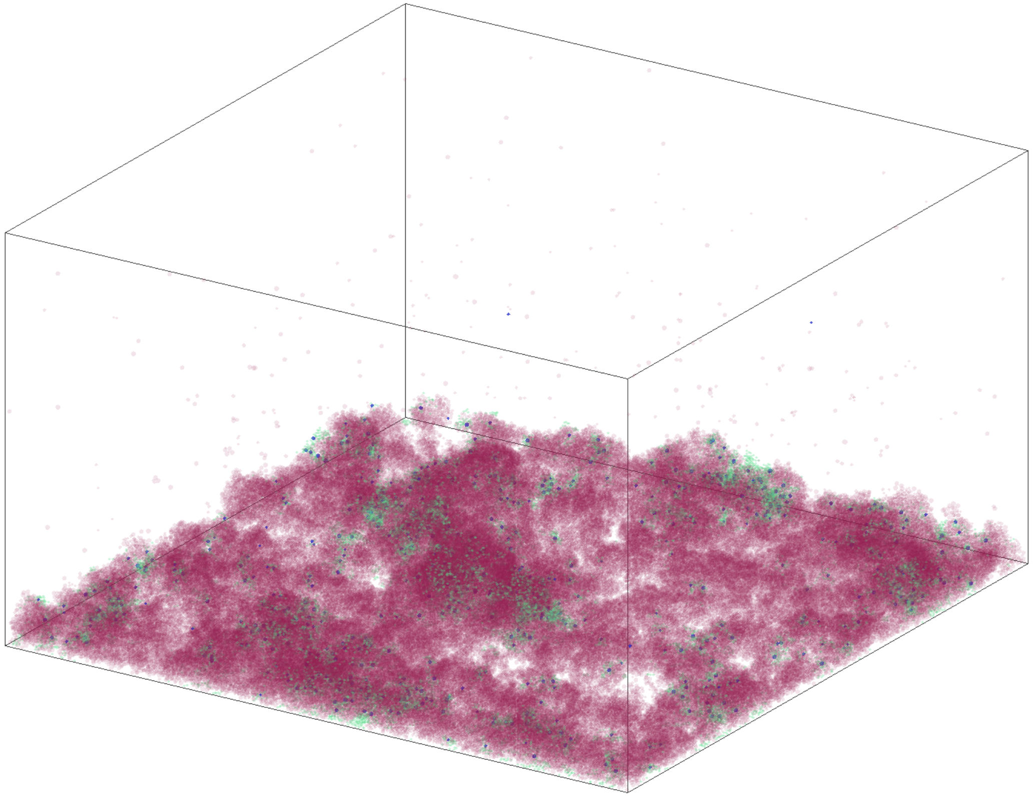
Changing strength of cell-cell interactions relative to cell-surface interactions leads different biofilm architectures (see slides 16 → 20)

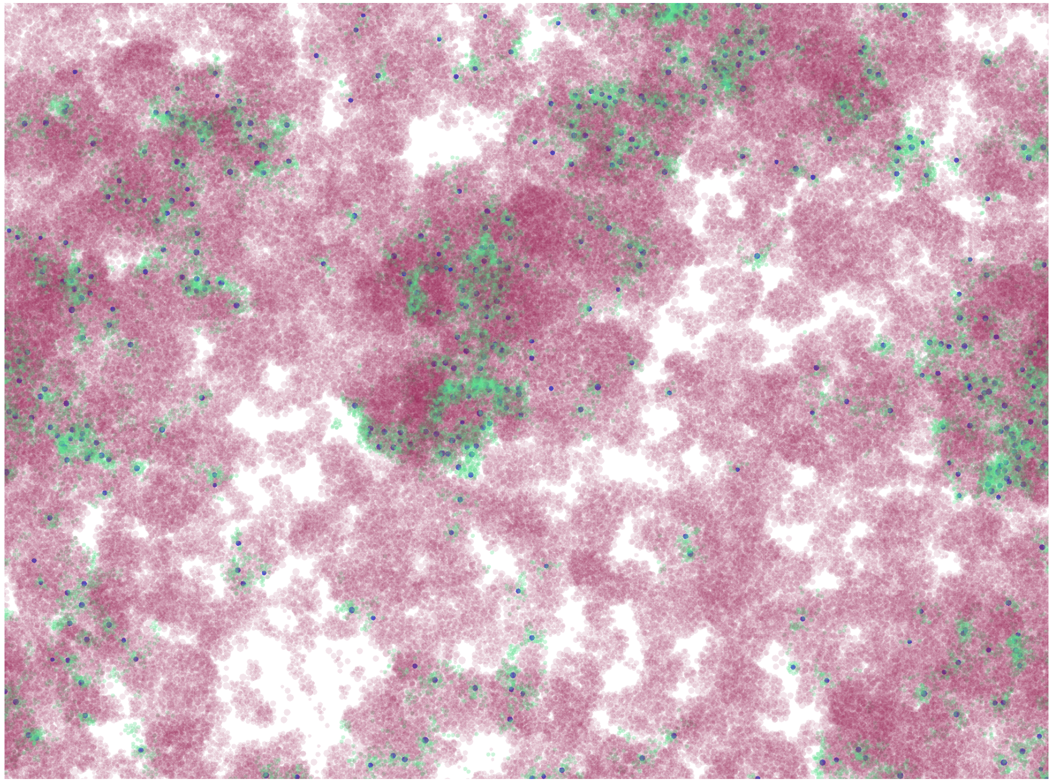




(A) shows biofilm from top

(B) shows biofilm from side

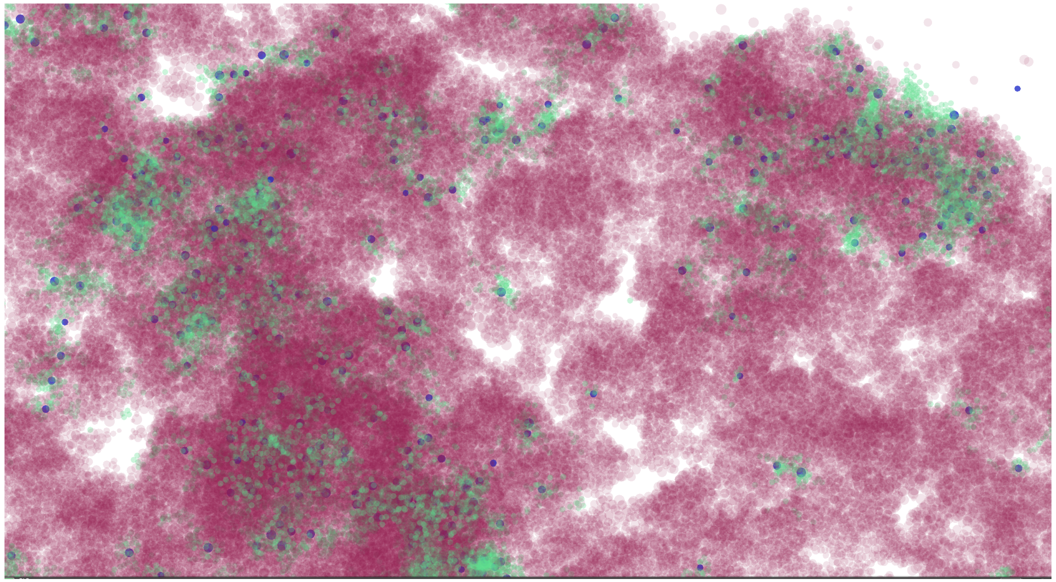




Arbitrary 3D dual species biofilm model

Modelled as planktonic cells which may adhere to the substratum, a large population of red cells and very few dark-blue cells.

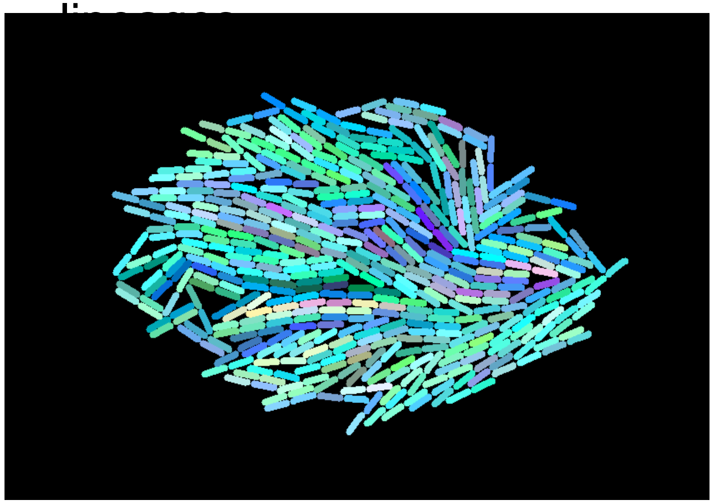
Red cells have a higher growth rate than the dark-blue cells

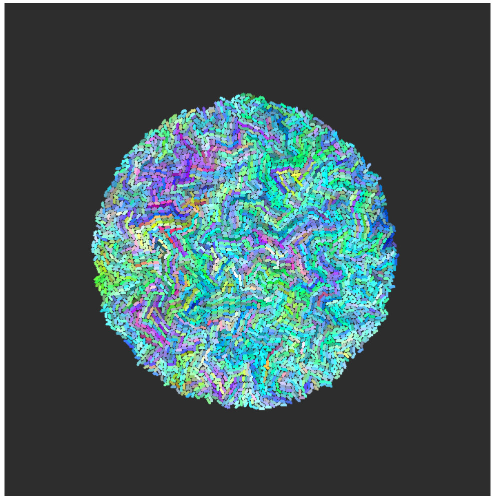


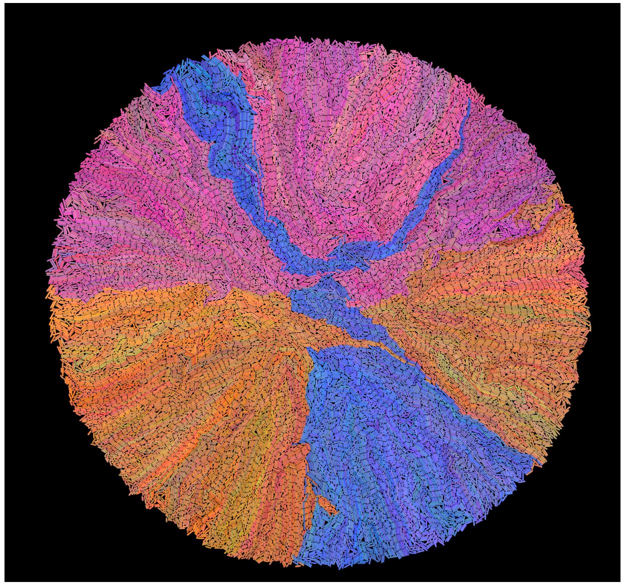
Dark-blue cells produce EPS (green) at a slow rate

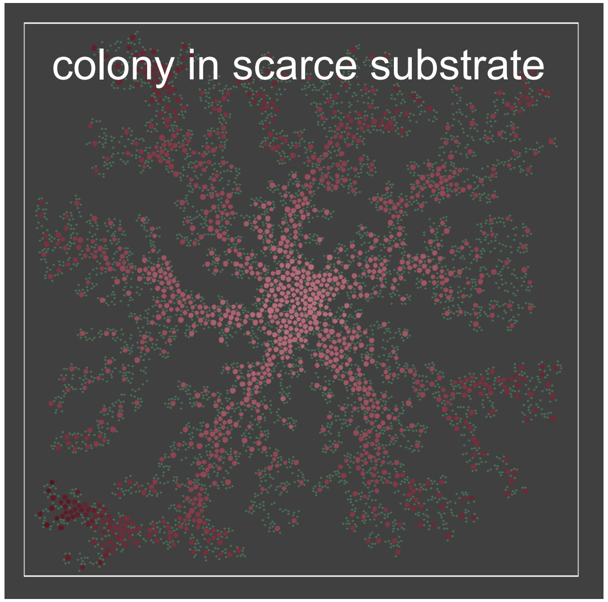
Simbiotics “Zoo”

2D rod-shaped cell colony growth displaying lineages

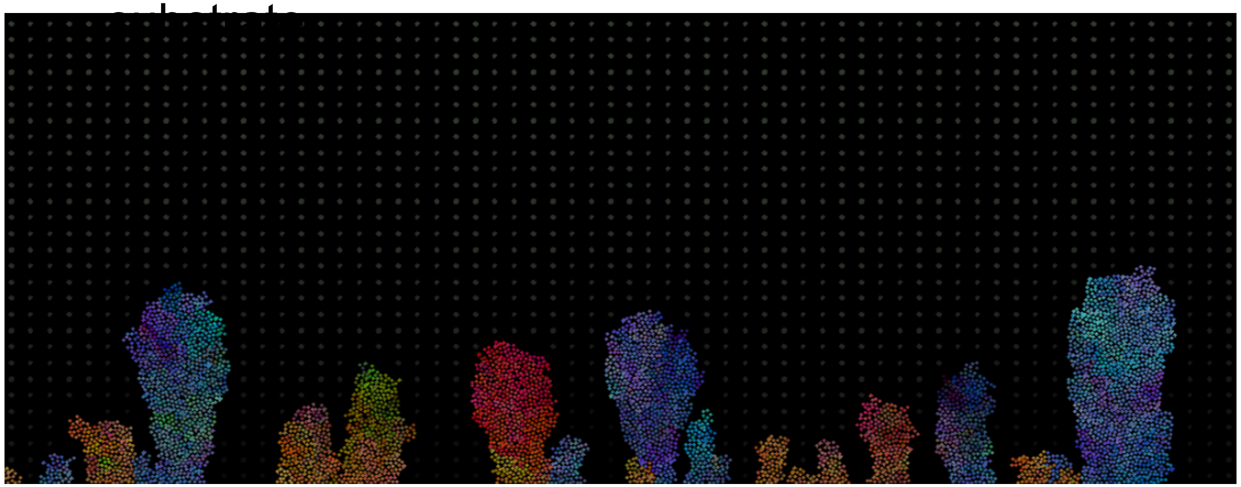






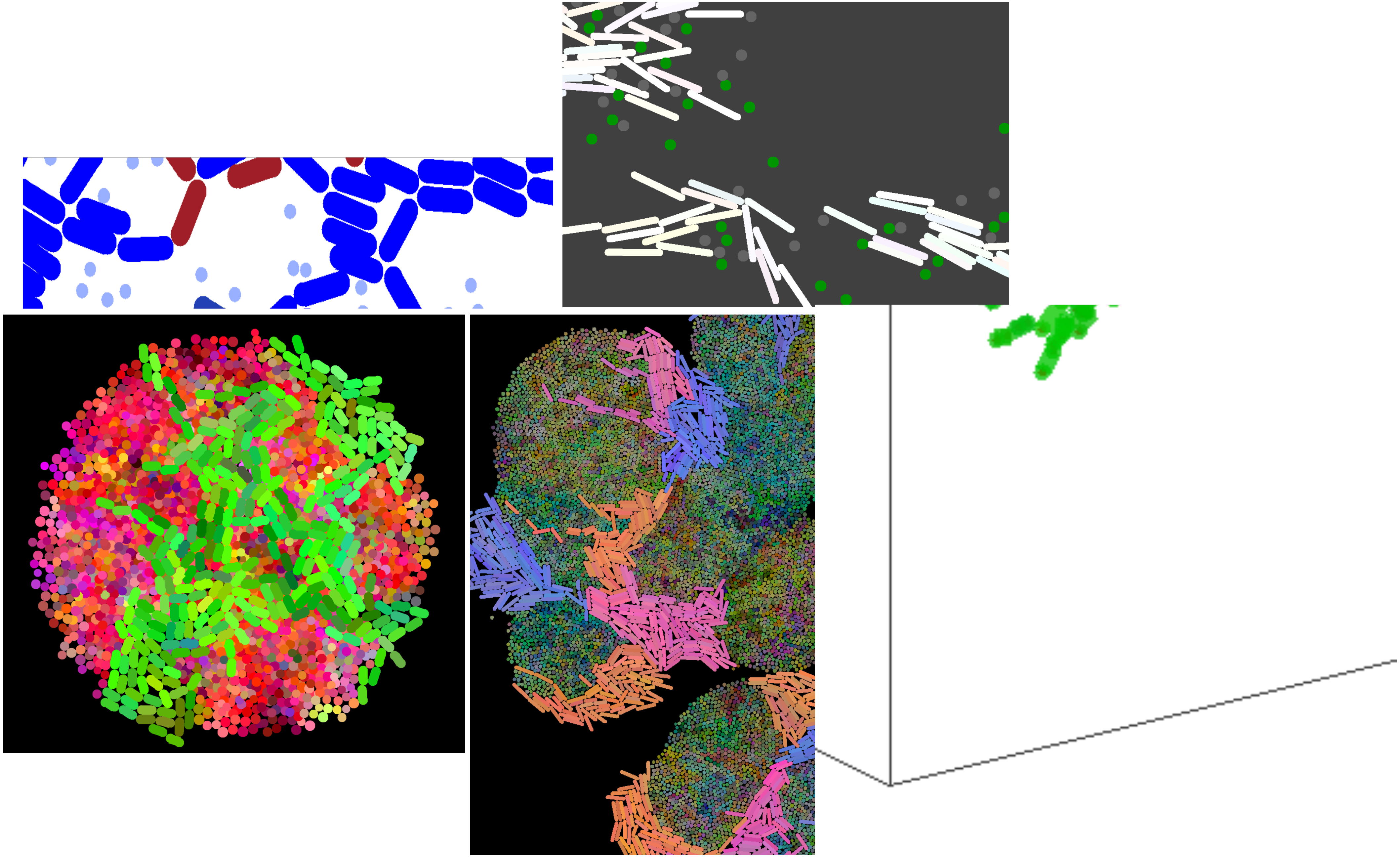


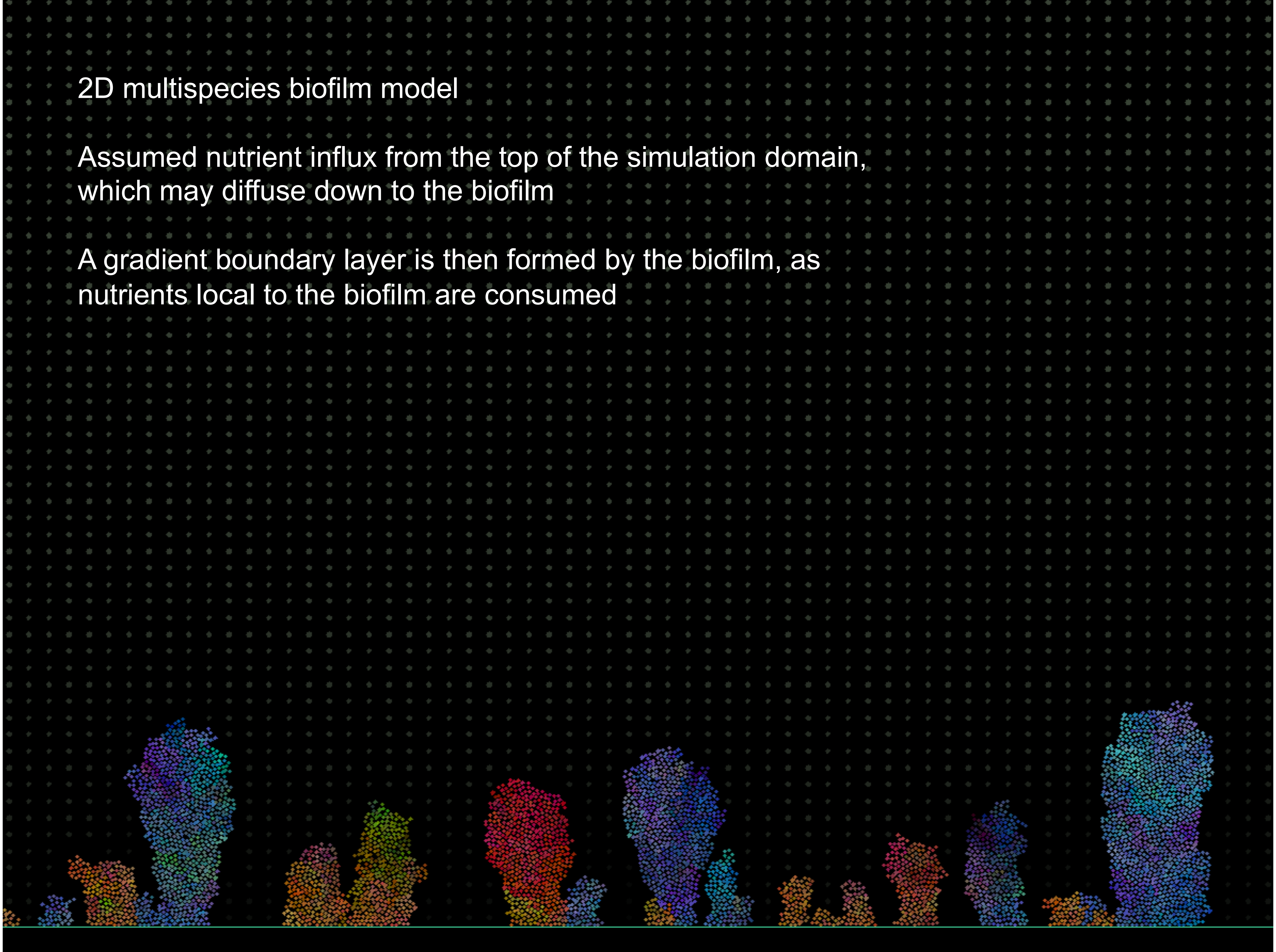
2D biofilm growth under scarce substrate



Simbiotics “Zoo”

Arbitrary simulations of mixed sphere and rod-shaped cells

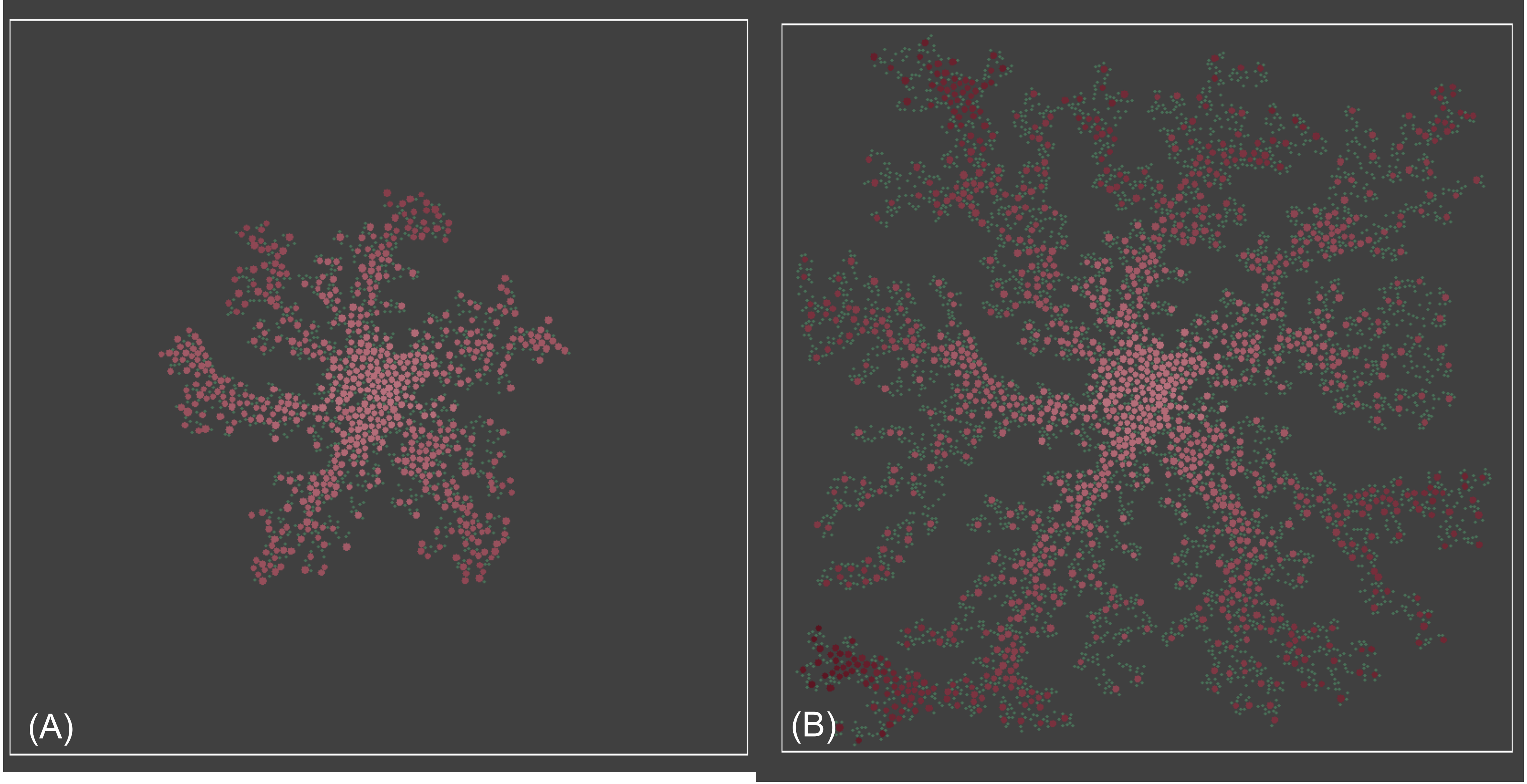




2D multispecies biofilm model

Assumed nutrient influx from the top of the simulation domain, which may diffuse down to the biofilm

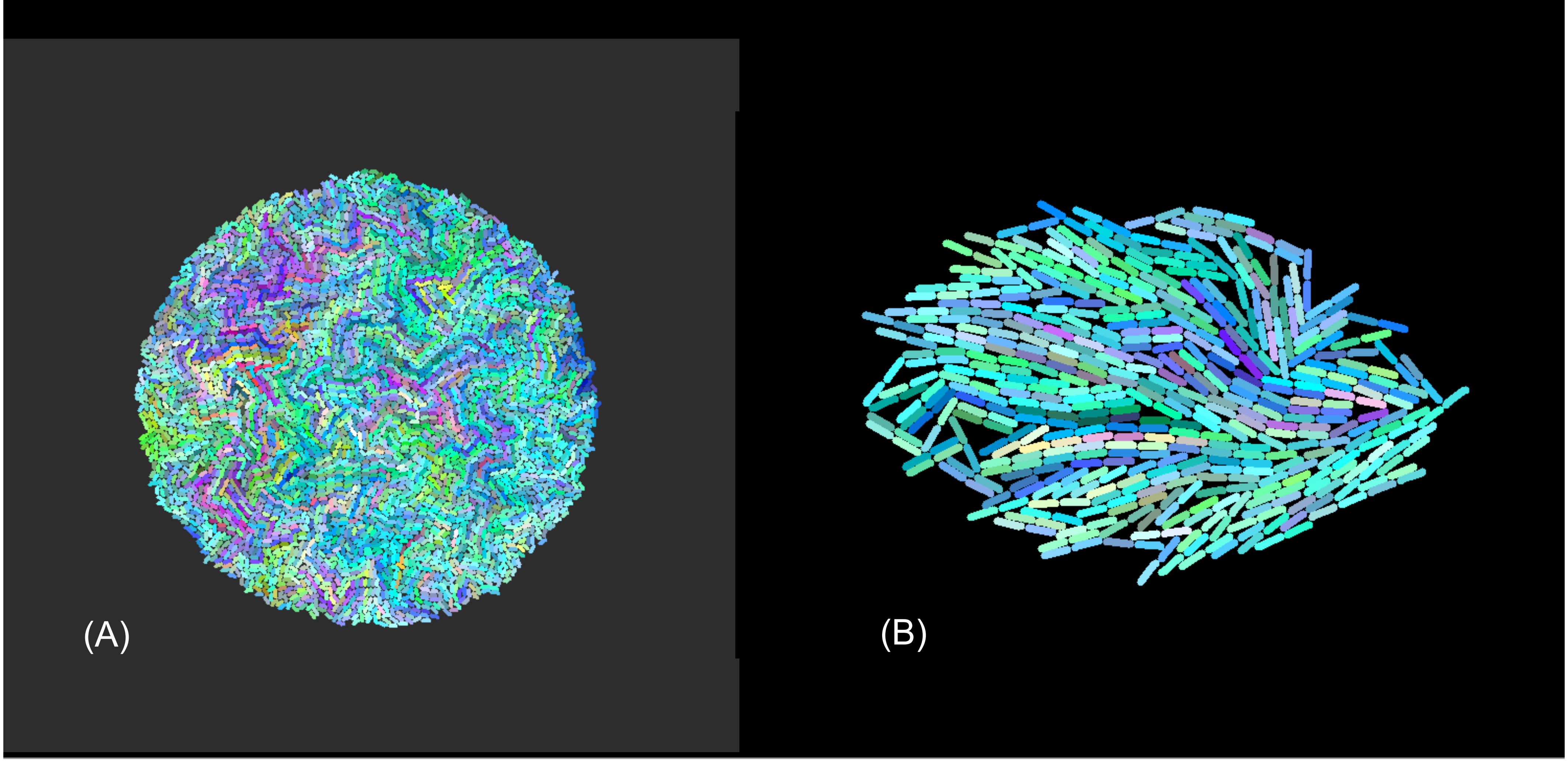
A gradient boundary layer is then formed by the biofilm, as nutrients local to the biofilm are consumed



2D colony development (prototype) – modelling nutrient dependence on a surface

Nutrients are initially evenly distributed across the substratum and have an low diffusion rate, they do not degrade

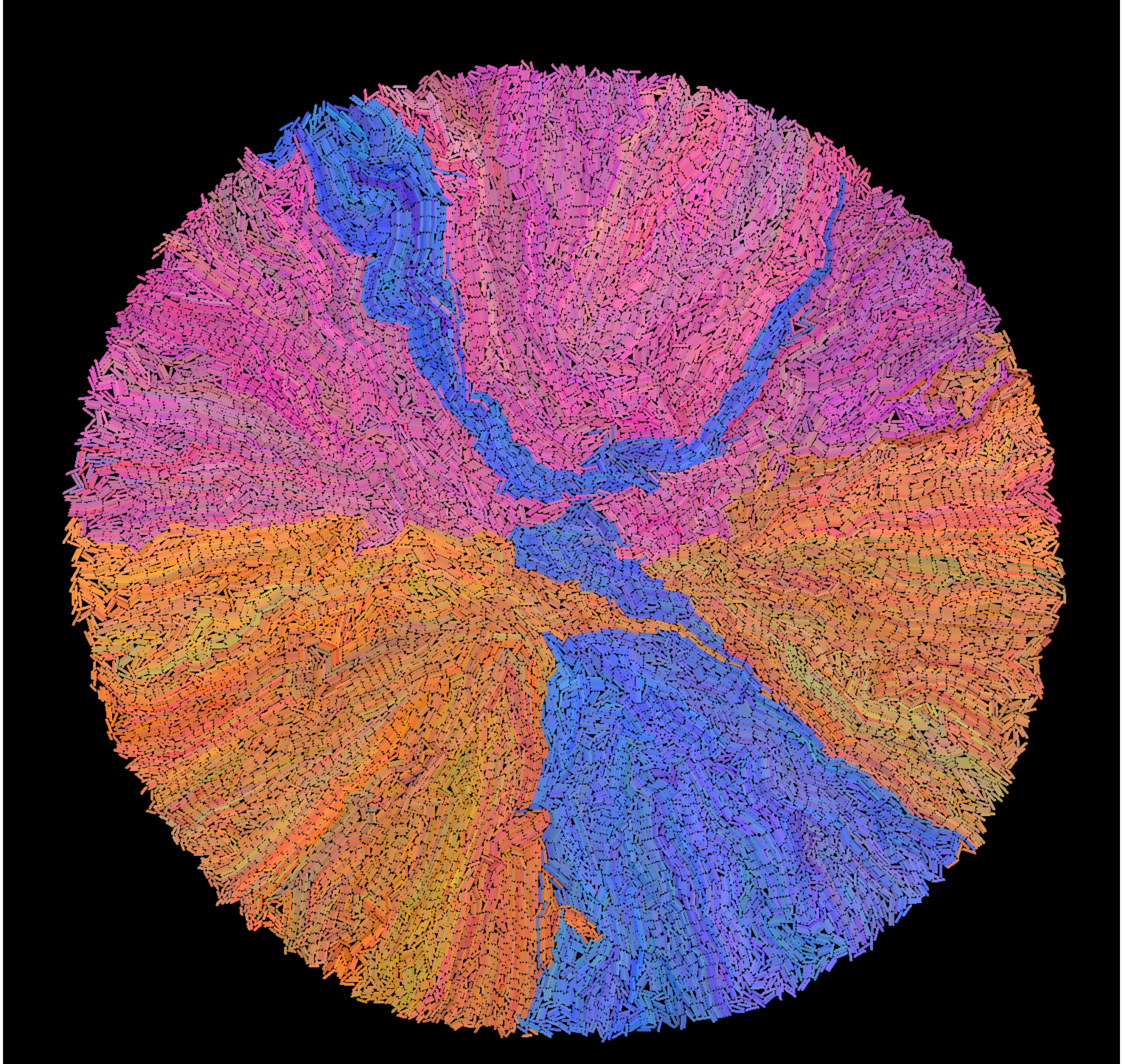
A single bacteria seeds the colony in the center – (A) and (B) are snapshots of colony at later time points



Single species colony of bacillus (rod) shaped bacteria

Both colonies grown from a single cell

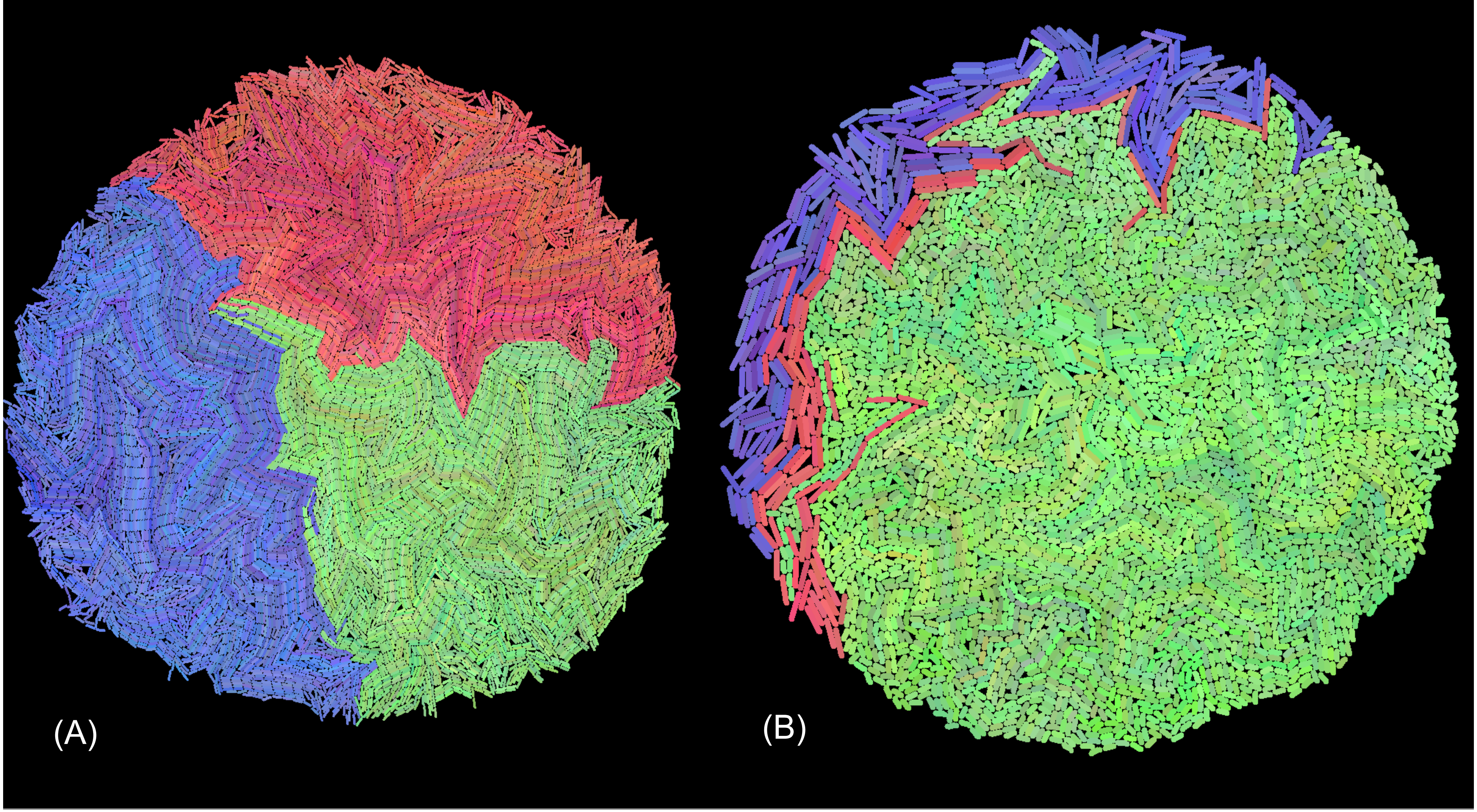
The species in (B) has a longer cell length than in (A)



Large colony of bacillus (rod) shaped bacteria

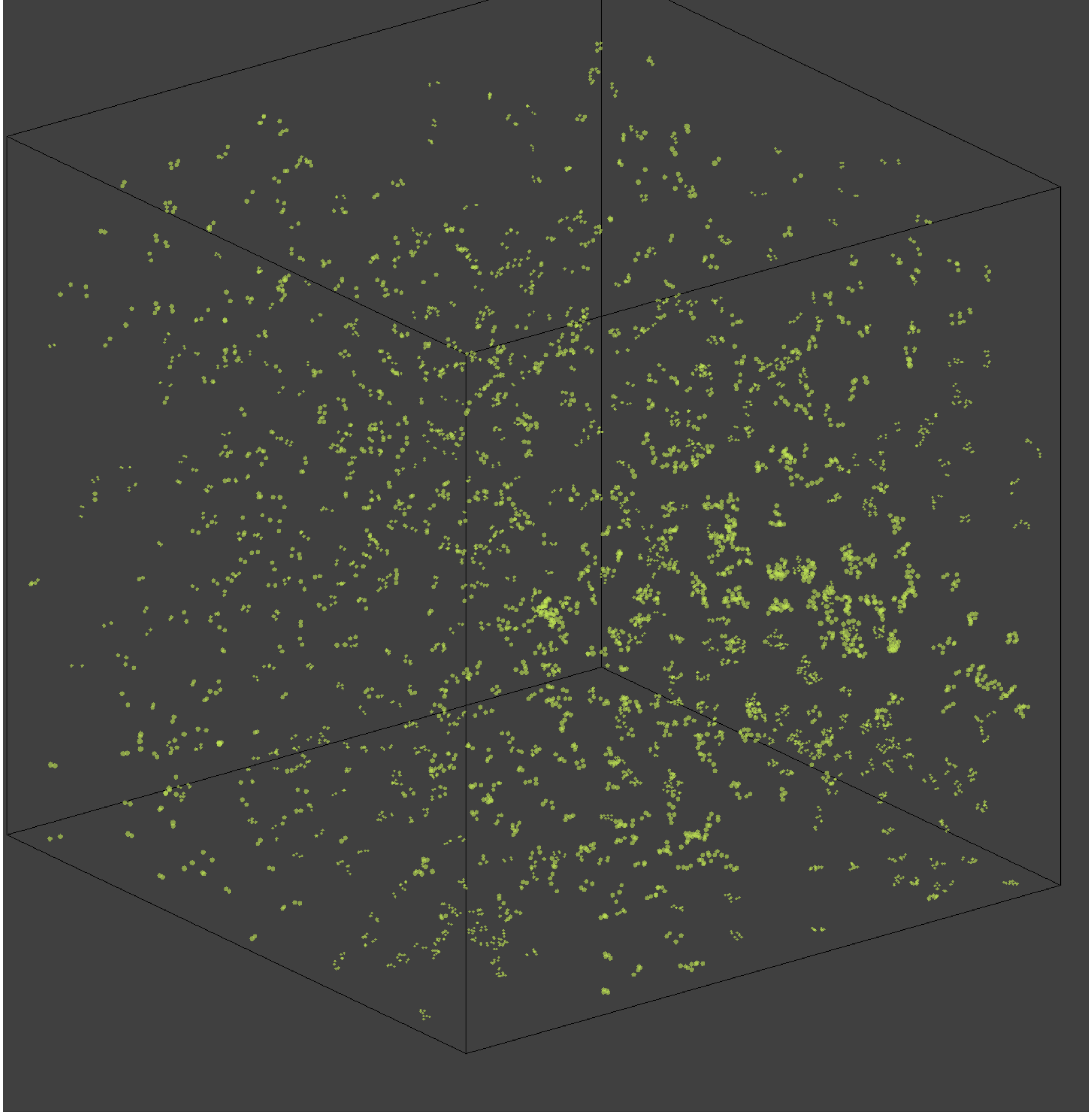
Three species, starting with one of each species as the initial condition

Assumed abundance of nutrients



2D multispecies colony development

A constant nutrient concentration is assumed. In (A) all 3 species have the same growth rate. In (B) The red and blue species have a longer cell length, however as their growth rate (gain in remains the same, they have a relatively longer generation time, sllowing the green species to dominate.



Growing planktonic in a nutrient limited environment

———

A well-mixed population which are only passively motile (they do not hunt nutrients) are suspended in a fluid

There's a constant influx of nutrients from a point, which may diffuse and degrade through the environment resulting nutrient gradients

FAQS

How do you design models in Simbiotics?

Currently Simbiotics models can be designed by attaching your SBML models to the default Simbiotics model, or by writing a custom Java model.

How many cell species can be defined/how are they defined?

The number of cell species is unbounded.
Species are designed by selecting a physical geometry for the species, and attaching submodels from the library which describe specific bacterial behaviour (they are parameterisable)

Common questions

How do populations of SBML models interact in Simbiotics?

Each cell has its own SBML model, the chemical species and concentrations in the SBML model are synchronised with the Simbiotics state.
Chemicals may diffuse out of the cell, through extracellular space, and to other cells/

Through this, two individual, spatially located SBML models may interact.

Common questions

How is cell adhesion modelled?

It is modelled by springs holding the cells together.
Springs may break if enough energy is applied to pull them apart.

The modeller may define surface structures on cell species, and define interactions between them.

A spring forms between two according to the following:
Upon the event of a cell-cell collision where the cells having a matching adhesin-receptor pair, there is a rate (probability per unit that) that the spring will form.

(note: it's the same for cell-substratum adhesion)

Common questions

Do you have 'process X' modelled in Simbiotics?

(If the answer is no, then this is the type of answer I tend to give)

That feature isn't currently present, however as Simbiotics is a framework which includes an extendable library, new processes can easily be built and added to the library.

How do you design new library modules/processes?
They are developed by creating a java class which implements or extends one of the simbiotics core interfaces/objects.
There is a tutorial in the user manual on how to do this.