BacArena: Simulation of Interactions in Microbial Communities using Genome-wide Metabolic Reconstructions

Johannes Zimmermann
Eugen Bauer

January 27, 2015

Abstract

Microbial communities are essential for global ecosystems and human health. Computational modeling of microbial consortia is thus a major goal in systems biology and microbial ecology.
BacArena is a project to simulate bacterial behaviour in communities. A lot of progress is done in the last years to gain genome wide metabolic reconstructions of certain organisms, which open a wide field of mathematical analysis. One of this new methods is flux balanced analysis (fba) to estimate optimal metabolic fluxes under certain constraints. By this work advanced models are possible, which are available in a defined, exchangeable format (*SBML*). The idea of this project is to use this existing reconstructions and put them in a spatial and temporal environment to study their possible interactions. This is achieved by the combination of agent based modeling with fba. Each bacterium is considered as an agent with individual states, own properties and rules to act. Agents are located on a grid where they can move and interact via metabolic exchanges computed by fba.
The starting point for our project is curiosity of what could be done with this huge models. We just throw those models into an arena to see what kind of actions will evolve.

Contents

1 Introduction

1.1 Microbial metabolic ecology

Microbial communities pose important roles in the cycle of matter [**Larsen12**] as well as human health and disease [**Aziz13**]. The comprehensive understanding of the interaction between microbes is thus a major goal in microbial ecology and systems biology.
Microbial consortia often consist of a diverse composition, which degrade complex compounds in multiple steps by different microbes [**Breznak94**]. This division of labour can be realized by the aggregation to biofilms in which multiple layers of microbes are associated with each other and interact by exchanging various metabolites [**Davey00**]. An example for the cross-feeding between microbes is the interspecies hydrogen transfer. Here, methanogenic archea are associated with bacteria or protists, which produce hydrogen after anearobic degradation [**Stams94**]. The hydrogen can be taken up as an essential substrate by the methogens, which profits the producer by the removal of the products and thus the thermodynamic limitation [**Stams94**]. Therefore both partners benefit from this interaction. A close spatial aggregation of the partners can further optimize their individual benefits, since the hydrogen can be exchanged faster.
Recent advances in systems biology made it possible to study the metabolic interactions of multiple species on the systems level [**Zomorrodi12**]. In particular constrained based modeling can be applied to model interspecies metabolic exchanges [**Klitgord10**].

1.2 Constrained based modeling

Constrained based modeling is a successfully applied method in systems biology ([**Esvelt2013**], [**Klipp2010**] p. 353), where the metabolism of single species is considered as a network of biochemical reaction. The reaction network itself can be represented more formally with differential equations using mass action kinetics. Because of the high numbers of reactions and metabolites the resulting system of equation and the solution space is high dimensional. Therefore, linear algebra is used for a simplified description:

$$\frac{dx}{dt} = S \cdot v$$

where $x \in \mathbb{R}^m$ is a vector consisting of concentrations of all m metabolites, $S \in \mathbb{Z}^{m \times r}$ is the stoichiometric matrix, which includes the net consumption/production of all r biochemical reactions and $v \in \mathbb{R}^r$ is the flux vector which contains in general nonlinear kinetic relationships.
Now several *constraints* could be applied to solve the problem more easily. The most prominent constraint is the equilibrium or steady state $dx/dt \stackrel{!}{=} 0$. It is a reasonable assumption for a metabolic model, because there is evidence for a metabolic steady state in general (i.e. no net change for every metabolites at each time point)([**Harris1995**] p. 10-11).
One important constrained based modeling approach is flux balance analysis (fba) ([**Varma1994**], [**Orth2010**]), which will be of utmost importance for this project. Here, the former non-linear problem $dx/dt = S \cdot v$ diminishes to $dx/dt = S \cdot v \stackrel{!}{=} 0$, which constitutes a normal linear equation systems. Nevertheless, there are far more reactions than metabolites ($r > m$), so that this linear reaction system is under-determined. That is why other constraints like flux limits are added. Flux limits are reaction limits, which narrow down each reaction to some intervall (e.g. irreversible reaction number i has a flux $v_i > 0$). By this, the solution space is reduced. If the biomass composition, non and growth

associated maintenance (NGAM/GAM) is known, it is possible to formulate an optimization problem:

$$\begin{aligned} \underset{v}{\text{maximize}} \quad & b(v) \\ \text{subject to} \quad & S \cdot v = 0 \\ & l_i < v_i < u_i \end{aligned}$$

where l_i and u_i are the lower and upper limits for reaction i and $b(v)$ is biomass function, which is going to be maximized with respect to a certain flux v. Thus, we search a vector v carrying quantitative values for all fluxes in the whole reaction system, so that a certain function (here biomass function) is optimal. Although constrained based modeling frameworks for studying species interactions exist [**Klitgord10**]), the complexity of microbial communities is still difficult to asses with this approaches.

1.3 Agent based modeling

Complexity theory is a part of system science since 1970s, where order is not longer considered as something given but made by itself. Moreover, order is producible as a surface phenomenon by a complex process, which is i) self organizing, ii) secures its autonomity and iii) proceeds far from an equilibrium ([**Cilliers2007**] p. 8-10).
According to John Holland, who introduced the important notion of an *agent*, a complex adaptive system (CAS) is defined as follows:

> ,,We will view CAS as systems composed of interacting agents described in terms of rules. These agents adapt by changing their rules as experience accumulates." ([**Holland1995**] p. 10)

In this modeling paradigm no general differential equation governs the macro behaviour. The parts of the system called agents are explicitly described by *rules* instead of a theory. This enables the possibility to model ,,microscopic" phenomenons, which give individual properties and defined information to the agents. ,,If-then rules" are heuristic and could depict relationships, where no mathematical description exists. Therefore, agents have individuality, live in a surrounding area (grid) with limited radius, so that only local interactions in the neighbourhood governed by rules are relevant to produce a global phenomenons.
From this microscopic actions the global organization is produced. New properties and behaviour could occur and this is noted by the slightly magical term *emergence* ([**Zimmermann2010**] p. 36-39).
Agent based modeling (abm) has been successfully applied in ecological studies to model the complexity of global behaviours by simple local interactions between species [**Kreft98**].

1.4 Aim of the project

The aim of this project is to combine for the first time constrained based modeling in an ecological context with abm to model interactions in microbial communities. In particular microbes are represented as agents which interact with their surrounding substrate concentrations stored in a grid. These interactions are realized with fba, which is used as a rule.
This framework will be used to represent common observations in microbial ecology such as syntrophy.

2 Methods

2.1 Model overview

To model the metabolic the interaction of multiple species populations, each individual was represented as an agent on a grid environment (Figure 1). The grid environment was composed of different

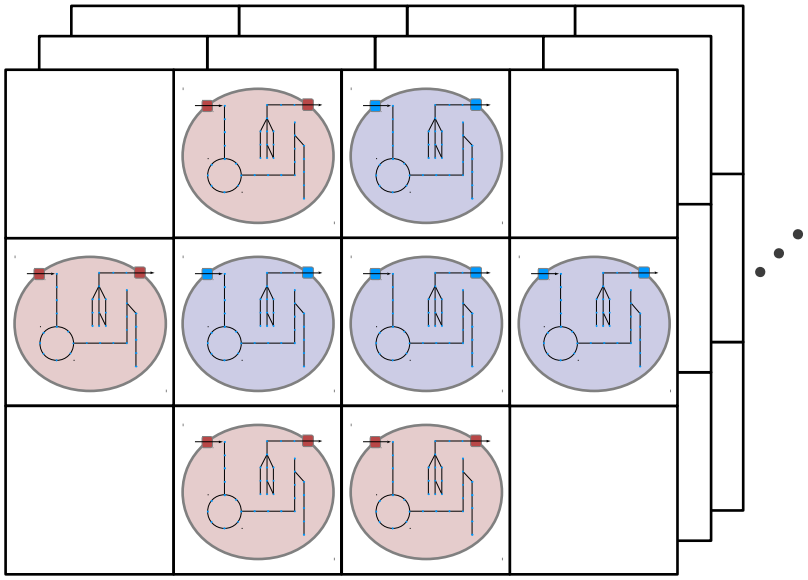


Figure 1: Overview of the grid environment with two different bacterial agent types (species) in red and blue. Each substrates has its own grid representation.

metabolite concentrations. Furthermore, the species type was recorded for each agent to call the respective genome-wide metabolic reconstruction in every iteration (Algorithm 1). The constraints for the subsequent fba were set according to the current metabolite concentrations of the grid cell, where the respective agent was located. The solution of the fba was used to adjust the biomass

Algorithm 1: Main model iterations called by `diffbac.R` with different functions applied to bacterial agents and metabolite.

```
for number of iterations do
    diffusion();
    for number of bacteria do
        fba();
        movement();
        growth();
    end
end
```

of each agent and to modify the metabolite concentrations according to the produced products and consumed substrates. This uptake and output of metabolites constituted the exchange with the virtual environment. A diffusion model was applied to spread the metabolite concentrations over the grid environment and a movement function allowed the random dispersal of each bacterial agent.

A central theme of `BacArena` was the modularity of each function and metabolic model to allow the extension and replacement of certain parts of the framework. Furthermore, the modularity also enables the inclusion of any desired amount of microbial species. In the following sections the modules of `BacArena` will be regarded more closely.

2.2 Representation

The main parts of the framework were implemented in the programming language `R` [**R12**]. Additionally, certain parts were implemented in `C++` and integrated in `R` with the package `Rcpp` [**dirk**].

2.2.1 Environment & Grid

Agents in `BacArena` were assigned to specific two-dimensional $n \times m$ grid positions $i, j \in \mathbb{N}$ and a type variable indicating the species. The grid is a discretization of space and could be imagined as a chess board, where the agents can move like chess pieces. One single part of the grid is called *cell* with no biological meaning.
For each grid positions certain metabolite concentrations were also recorded and stored in a separate matrix. For each metabolite a own matrix was constructed (Figure 1). The bacterial agents could interact with this environment by the consumption and production of metabolite concentrations.
Regarding the grid environment continuous, boundary conditions were chosen, i.e. the rectangle grid is forming the surface of a torus/donut (horn-torus in the case of a square grid).

2.2.2 Bacteria

Bacterial populations were represented as a matrix, which had four columns and rows according to the current number of agents on the grid. The first two columns contained the discrete positions of the agents on the grid. The third column indicated the species type of the respective agent. The fourth column stored the current biomass value of the bacteria.
For certain bacterial species, published genome-wide metabolic reconstruction were used as a representation of the individual metabolism by performing flux balance analysis. In the present study the *SBML* models for *Escherichia coli* (iAF1260) [**Feist07**], *Methanosarcina barkeri* (iAF692) [**Feist06**] and *Clostridium beijerinckii* (iCM925) [**Milne11**] were used. For each model the metabolites of the published artificial minimal medium (except the respective essential carbon and electron sources) were set to concentrations *ad libitum*. The biomass function for subsequent fba was adopted from the respective *SBML* models.
According to experimental studies the metabolite fluxes as well as the growth (GAM) and non growth associated maintenance (NGAM) of the models were set to biologically relevant constraints. Those adjusted flux constraints for each model and the respective reference can be found in Table 1.

Table 1: Flux constraints as well as the growth (GAM) and non growth associated maintenance (NGAM) set for each model according to the respective reference. Values are given in $\frac{\text{mmol}}{\text{h}}$.

	E. coli	*M. barkeri*	*C. beijerinckii*	Reference
NGAM	8.39	1.75	8.5	[**Feist07**], [**Feist06**], [**Milne11**]
GAM	59.81	30	40	[**Feist07**], [**Feist06**], [**Milne11**]
Glucose	11	–	9.39	[**Feist07**], [**Milne11**]
Oxygen	18.2	–	–	[**Feist07**]
Succinate	16	–	–	[**Orth11**]
Methanol	–	16	–	[**Feist06**]
Hydrogen	–	41	–	[**Feist06**]
Pyruvate	–	5	–	[**Feist06**]
Acetate	–	8	3.41	[**Feist06**], [**Milne11**]

In each iteration in `diffbac.R` (Algorithm 1), rows were deleted and included according to the growth function (see below). The grid positions were updated according to the movement function and the solution of the fba was added to the current biomass value for each agent.

2.2.3 Substrate

Metabolite concentrations were represented as matrices $m_s \in \mathbb{R}^{n \times m}$ for all substrates s. According to the requirements of the used Algorithm 1genome-wide metabolic models certain metabolites were selected. In the present study these metabolites were: acetate, aketoglutarate, carbon dioxide, ethanol, formate, fumarate, glucose, water, proton, lactate, oxygen, phosphate, pyruvate, succinate, hydrogen, methanol, methane, acetone, butyrate and butanol. This list can be extended easily by adding another substrate matrix of an arbitrary metabolite with the definition of additional exchange reactions.
According to the simulations of diverse metabolic phenotypes, the matrices of certain metabolites were initialized with a concentration of 70 mmol per grid cell, whereas the remaining metabolites were initialized with 0 mmol.

2.3 Interactions as rules

The different representations were closely integrated with each other using rules, which acted on the level of each individual.

2.3.1 Movement

To allow the dispersion of the microbial populations, a movement model was implemented (Algorithm 2). Here, each individual agent chooses a random position in the von Neumann neighbourhood,

Algorithm 2: Movement of bacterial agents in the von Neumann neighbourhood with i and j as the current positions on the grid.

$a \leftarrow i + random(1, -1)$;
$b \leftarrow j + random(1, -1)$;
for *number of bacteria* **do**
  if $(a,b) \in$ *bacteria positions* **then**
    do not move
  else
    $i \leftarrow a$;
    $j \leftarrow b$;
  end
end

if this cell is not occupied by another microbe. The current two-dimensional grid position is then overwritten with the position of the neighbour. In case the position is located on the boundaries of the grid, the cell on the opposite side of the grid is chosen as a neighbour.

2.3.2 Diffusion

For the diffusion of the different metabolites a naive diffusion model was implemented (Algorithm 3). In this strategy, the states of randomly chosen neighbouring cells is interchanged with asynchronous updates. The interchange is implemented by replacing the individual values with the concentration mean of the selected cells.

2.3.3 Flux balance analysis

We used fba to represent the metabolic requirements of each microbe in the grid environment. As described in chapter 1.2 fba determines a flux vector for all reactions by maximizing (in our case) the

Algorithm 3: Diffusion implemented in c++ and included in R using Rcpp [**dirk**]

```
for all substrates s do
    l ← 0;
    n ← rows;
    m ← columns;
    while l++ < n · m do
        i ← random(1, n);
        j ← random(1, m);
        min ← random local minimum in neighbourhood of cell i, j;
        m ← (s(min_i, min_j) + s(i, j))/2;
        s(i, j) ← m;
        s(min_i, min_j) ← m;
    end
end
```

biomass function. To solve this optimization problem we used *lpsolve* [**lpsolve**], which is available as a *R* package [**Rlpsolve**]. The package *lpsolve* is a mixed integer linear programming solver, which is ,,based on the revised simplex method and the branch-and-bound method for integers"[**lpsolvedocu**]. The respective species specific *SBML* model was loaded for each organism located on the grid (Algorithm 1). In the subsequent fba, several constrains were applied:

1. Lower bound of irreversible reaction fluxes is set to 0
2. All internal metabolites were assumed to be in flux equilibrium, i.e. $S \cdot v = 0$.
3. Available substrate concentrations control the flux of all exchange reactions (lower bound were set to metabolite concentrations)
4. Certain empirically validated flux boundaries (Table 1) limits exchange reactions even further.

Linear optimization delivers a steady state flux vector v, which maximizes the used biomass function. Furthermore, the fluxes were multiplied with the current biomass (accumulated growth) of each microbial agent, to represent the unit of v as $[\frac{\text{mmol}}{\text{g}_{\text{DW}} \cdot \text{h}}]$ (higher biomass per dryweight $\sim$ higher fluxes). The solution flux vector of the fba optimization was taken to redefine the metabolite concentrations on the grid by simple arithmetic operations. Here, all uptake exchange reactions lower the available substrate concentrations in each time step (1 h) by subtracting the present concentrations with the concentrations taken by the microbe. Conversely, the providing exchange reactions raise the product concentrations in each time step (1 h) by adding the concentrations produced by the microbe to the present concentrations.
To represent the biomass the individually determined growth rates $[\frac{\text{mmol}}{\text{g}_{\text{DW}} \cdot \text{h}}]$, defined by the flux through the biomass function according to the available substrates, were added to the existing biomass of each microbe. Therefore, the biomass accumulated in each time step (1 h).

Hashing As a possibility to speed up fba computations, we implemented a hashing function. We set up a table with *md5* hashes connected with the fba results as entries. By this ,,calculatory memory" it was possible to use former calculations again, to avoid reoccurring identical linear optimization cases.

2.3.4 Growth

Microbial growth was considered as the output of the fba-optimized biomass function. In case the fba did not find a solution, we considered a starvation phase, in which biomass was consumed. To produce population dynamics, two different processes were part of the model:

1. **Duplication**: Biomass accumulation allows the microbial reproduction (duplication).
2. **Death**: Starvation consumes the available biomass and leads to ultimate cell death.

Duplication The growth (GAM) and non growth associated maintenance (NGAM) was adopted from the literature individual microbe species (Table 1). NGAM, accounts for the obligatory ,,housekeeping" energy consumption of each cell, whereas GAM represents the energy consumption necessary for replication.
In `BacArena` all microbe agents start with a initial biomass of 1 symbolizing an intact cell. If the accumulated biomass in the growth model reaches the double amount of the initial biomass, duplication was enabled. The daughter cell was randomly placed in the direct neighbourhood of the parent, similarly to Algorithm 2. In this process the current biomass was equally divided between daughter and parent cell. If the chosen grid position is occupied by another microbe the daughter cell was removed (death).

Death Flux balance analysis considers a fixed NGAM flux and growth depending GAM part of the biomass function. Under starving conditions, where no fluxes according to fba calculations can be found, we considered the current biomass as a reservoir, from which the cell can ,,feed" on. The biomass was subsequently reduced with

$$\text{biomass}_{t+1} = -\frac{\text{NGAM}}{\text{GAM}} \cdot \text{biomass}_t + \text{biomass}_t$$

which is dependent on the adopted NGAM and GAM values of each individual *SBML* model (Table 1). The microbe agents were removed from the grid environment (death), if their current biomass was lower than the arbitrary threshold of 0.

3 Results

3.1 General results of the framework

3.1.1 Movement and Diffusion

The movement of the microbial agents showed the random dispersal of the microbes over the grid environment (Figure 2). The same was also observed in the diffusion model, where the metabolite flowed to the lowest concentrations on the grid (Figure 3).

3.1.2 Time consumption

Considering the time consumption of the whole framework (Figure 4), the calculation of the flux balance analysis required the highest amount of time. Furthermore, the overall fba calculation time was dependent on the number of microbe individuals on the grid, whereat higher numbers of agents resulted in higher overall calculation times. The same was true for the movement function, which consumed after the fba the highest amount of time. The time consumption of diffusion function

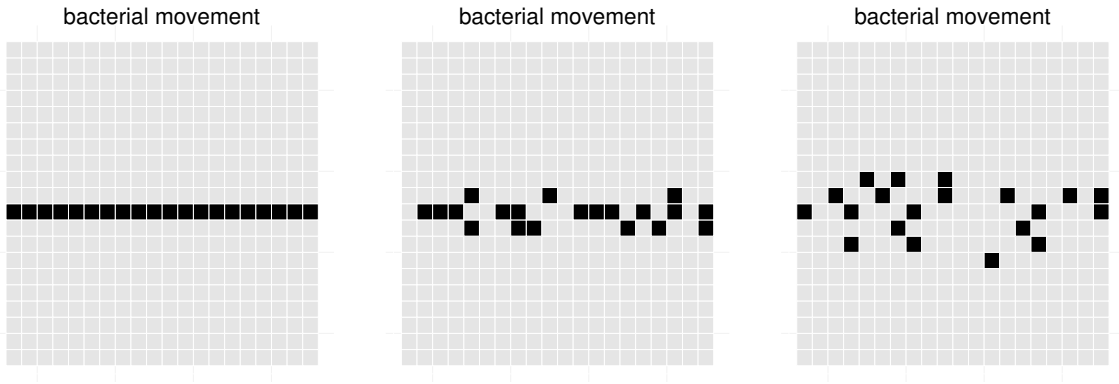


Figure 2: Bacterial movement starting with a line of bacteria in the middle of a 20×20 grid. Three different iteration steps are displayed (time step 1, 2 and 5).

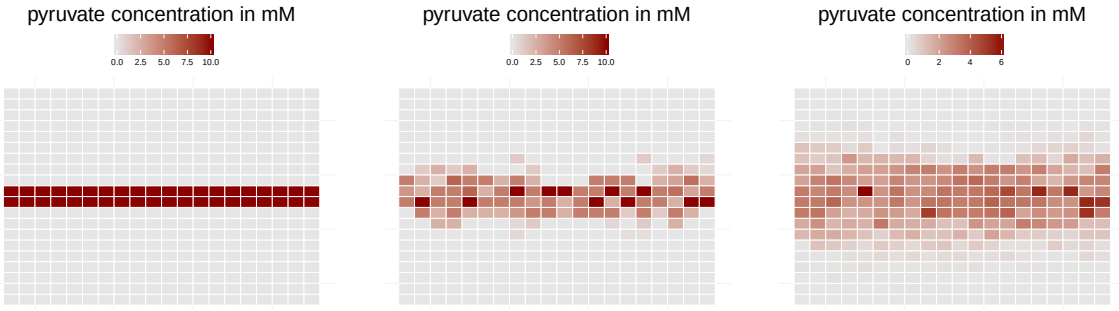
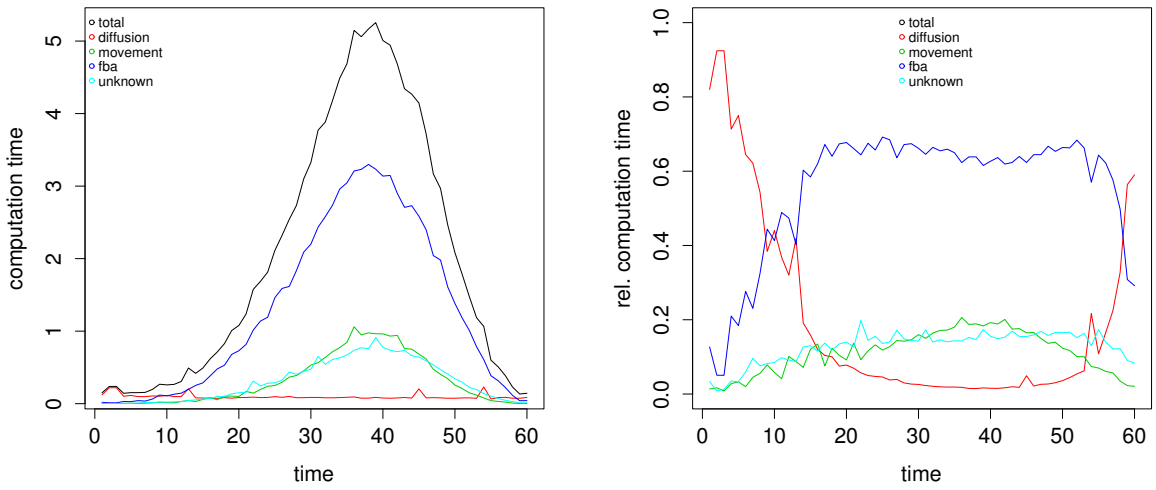


Figure 3: Diffusion starting with 10 mmol pyruvate in the middle of a 20×20 grid. Three different iteration steps are displayed (time step 1, 2 and 5).



A B

Figure 4: Absolute (A) and relative (B) time consumption over the simulation of the *E. coli* core model on a 20×20 grid.

was considerably lower than the other functions and constant over the whole simulation, since it is independent on the number of microbes on the grid. Other calculations for diverse matrix operations (primary *x*apply code in *R*) required with respect to the movement function a comparable amount of

time. Grid size was the dominant factor, because lager space increased the needed time of all parts. For larger *SBML* models the fba calculations consumed more time. The time consumption of the other function was independent of the used model and negligible for *SBML* models with a stochiometric matrix larger than 500×500 (reactions $\times$ metabolites). Therefore, fba was the time limiting factor of the simulations.

3.2 Population models

3.2.1 Escherichia coli core

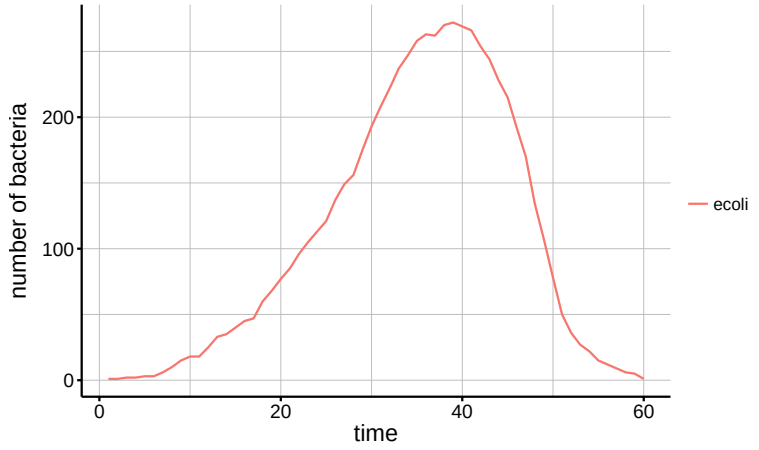
The *E. coli* core model was subjected to initial concentrations of the substrates glucose and oxygen to generate a population model and monitor the production/consumption of various metabolites (Figure 5). In the first time steps oxygen and glucose were consumed and CO_2 was produced. Additionally, fermentation products such as acetate, formate and ethanol were produced, which increased in concentration during the exponential phase of microbial growth. Acetate and formate were produced with high levels (average grid cell concentration of $> 60\,mmol$). The doubling time in the exponential phase was estimated as 7 iteration (h). The population growth reached the stationary phase after 30 iterations (h), after all substrates were consumed. In the subsequent death phase no metabolites were produced or consumed. According the dispersion of the microbes on the grid environment, substrates were consumed and metabolites produced (Figure 6).

3.2.2 Large Escherichia coli model (,,Bcoli")

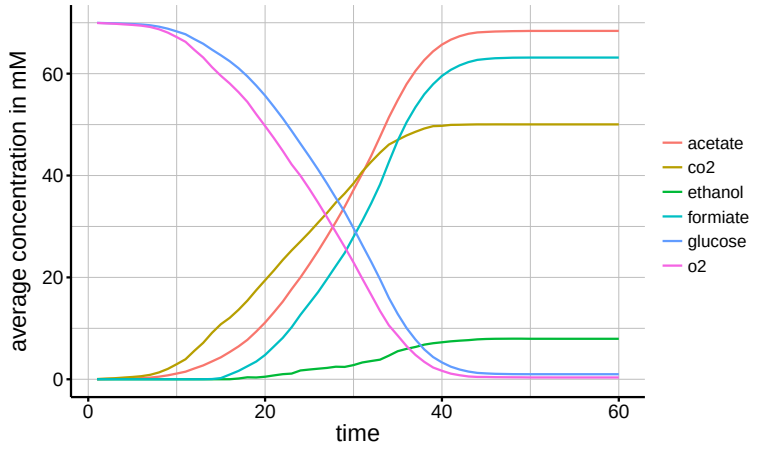
The *E. coli* model was subjected to initial concentrations of the substrates glucose and oxygen to generate a population model and monitor the production/consumption of various metabolites (Figure 7). In the first time steps oxygen and glucose were consumed and CO_2 was produced. More oxygen than glucose was consumed in the first iterations. Additionally, few time steps later fermentation products such as acetate and ethanol were produced, which increased in concentration during the exponential phase of microbial growth. CO_2 reached average grid cell concentration of $\approx 120\,mmol$, acetate $\approx 50\,mmol$ and ethanol had a final concentration of $\approx 10\,mmol$. In the simulation, no production of formate was found. The doubling time in the exponential phase was estimated as 6 iteration (h). The population growth reached the stationary phase in approximately 40 iterations (h), after all substrates were consumed. In the subsequent death phase no metabolites were produced or consumed. The population died out after 60 iterations. According the dispersion of the microbes on the grid environment, substrates were consumed and metabolites produced (Figure 8). Here, CO_2 was produced on every position the bacterial agents visited and acetate was preferably produced in the centre of the grid.

3.2.3 Methanosarcina barkeri

The *M. barkeri* model was subjected to initial concentrations of methanol as a substrate, to generate a population model and monitor the production/consumption of various metabolites (Figure 9). In the first time steps methanol was consumed and CO_2, methane and water was produced. Compared to water and methane, the production of CO_2 was considerably lower. The production of water and methane was similar. The doubling time in the exponential phase was estimated as 17 iteration (h). The population growth reached the stationary phase in approximately 80 iterations (h), after methanol was consumed. In the subsequent death phase no metabolites were produced or consumed. The population died out after 150 iterations.



A



B

Figure 5: Population dynamics of the *E. coli* core model on a 20×20 grid, with bacterial growth (A) and consumption/production of various metabolites (B). An initial concentration of 70 mmol per grid cell of glucose and oxygen was added to the environment. The seed of the random number generator was set to 55.

According the dispersion of the microbes on the grid environment, substrates were consumed and metabolites produced (Figure 10). Here, methane, CO_2 and water was produced and methanol consumed on every position the microbial agents visited.

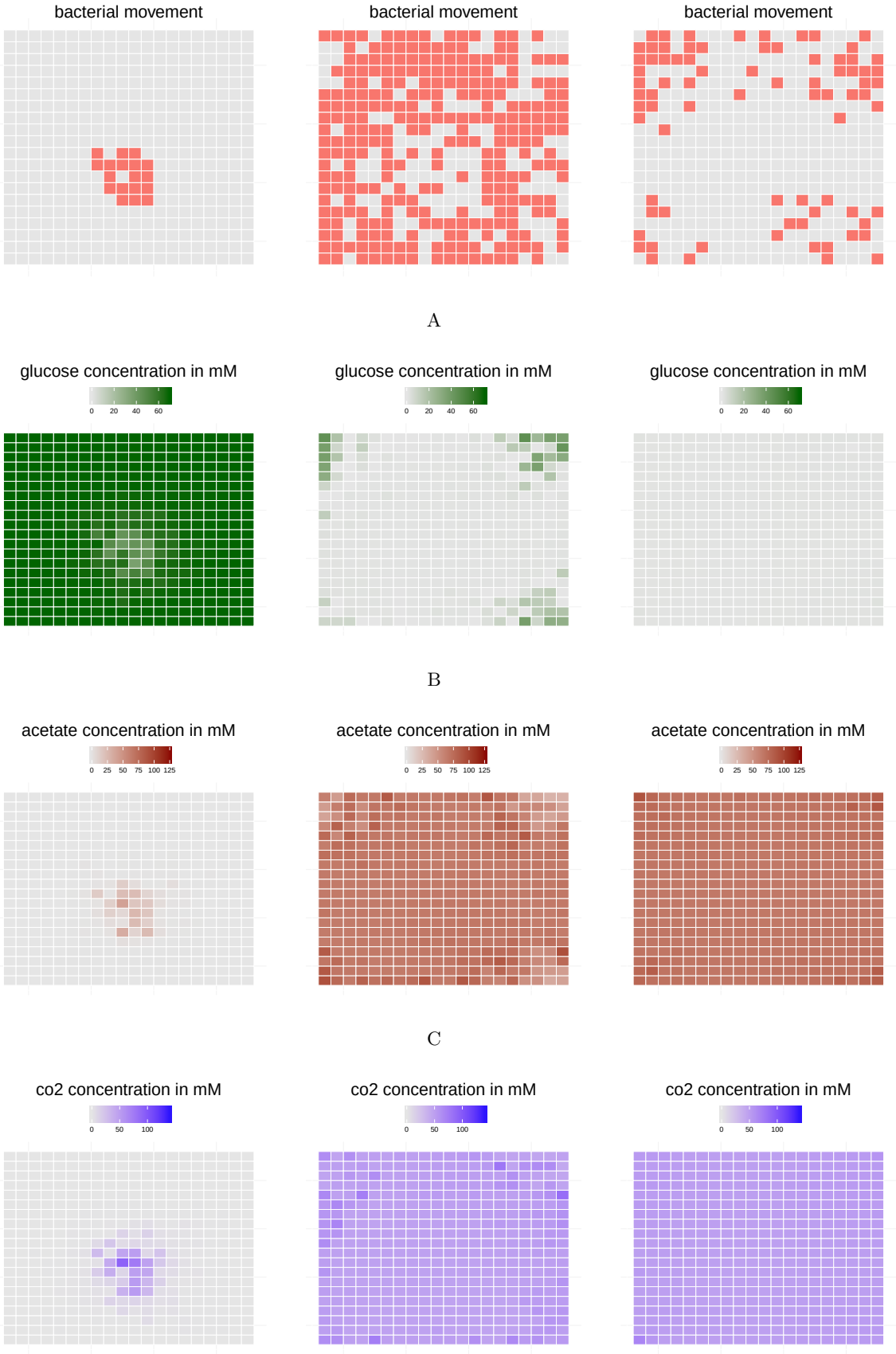


Figure 6: Population dynamics of the *E. coli* core model on a 20×20 grid, with bacterial movement (A) and concentrations of glucose (B), acetate (C) and CO_2 (D) (of time step 10, 40 and 50). The seed of the random number generator was set to 55.

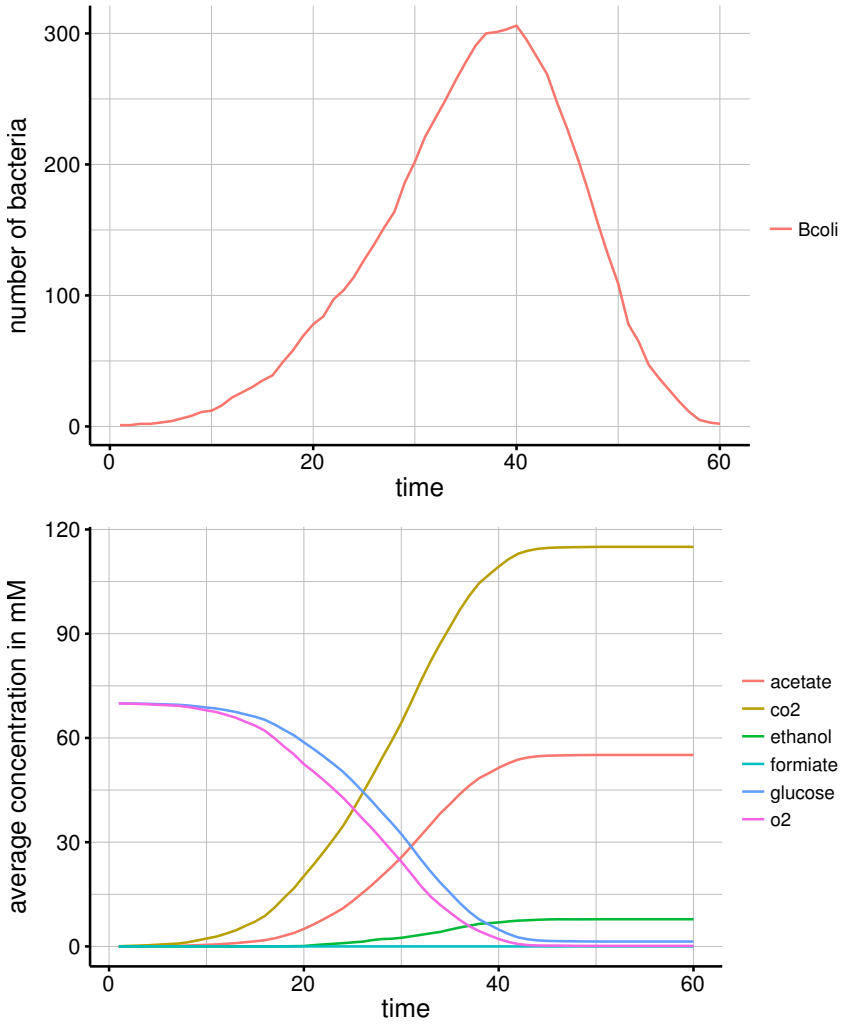
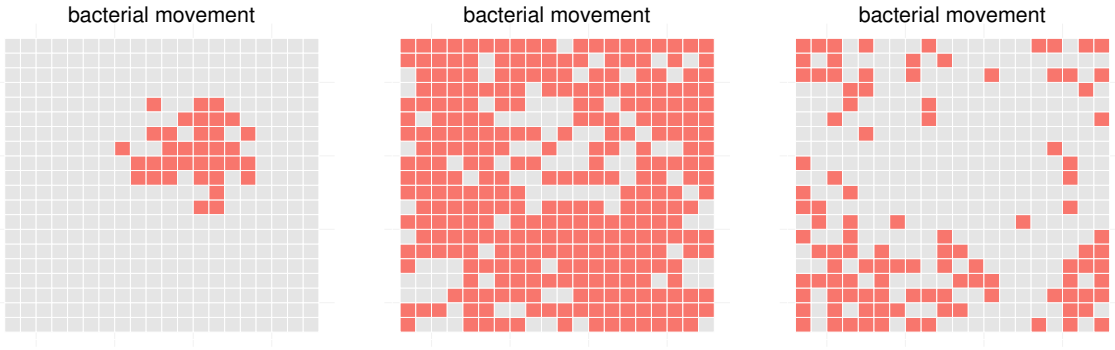


Figure 7: Population dynamics of the big *E. coli* model on a 20×20 grid, with bacterial growth (A) and consumption/production of various metabolites (B). An initial concentration of 70 mmol per grid cell of glucose and oxygen was added to the environment. The seed of the random number generator was set to 911.

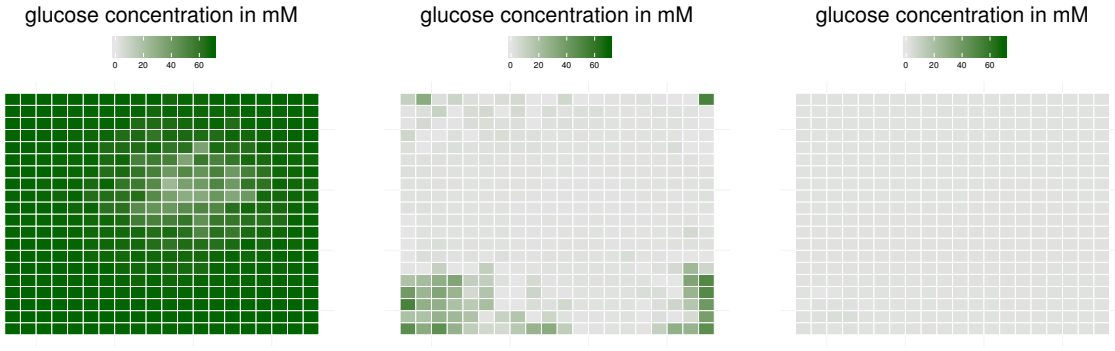
3.2.4 Clostridium beijerinckii

The *C. beijerinckii* model was subjected to initial concentrations of glucose as a substrate, to generate a population model and monitor the production/consumption of various metabolites (Figure 11). In the first time steps glucose was consumed and hydrogen, $C0_2$, acetate, butyrate and succinate were produced in increasing amounts. Compared to the other products hydrogen and CO_2 were produced much more. The doubling time in the exponential phase was estimated as 6 iteration (h). The population growth reached the stationary phase in approximately 45 iterations (h), after glucose was almost consumed. In the subsequent death phase no metabolites were produced or consumed. The population died out after 65 iterations.

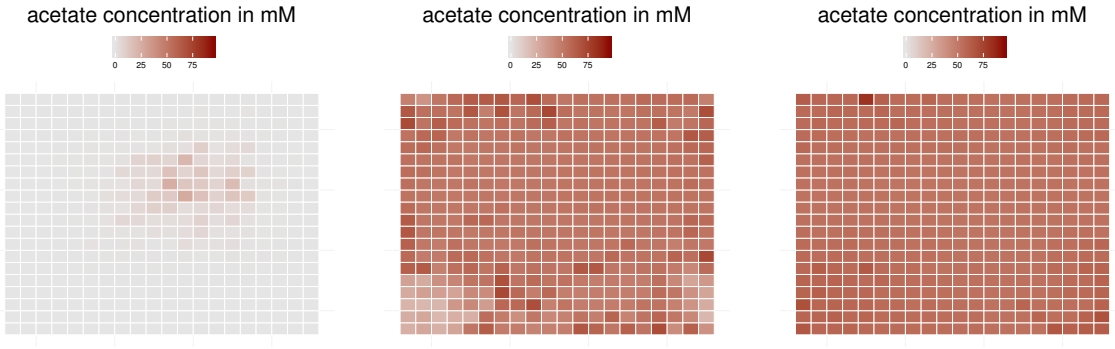
According the dispersion of the microbes on the grid environment, substrates were consumed and metabolites produced (Figure 12). Here, hydrogen, $C0_2$, acetate, butyrate and succinate was produced and glucose consumed on every position the bacterial agents visited.



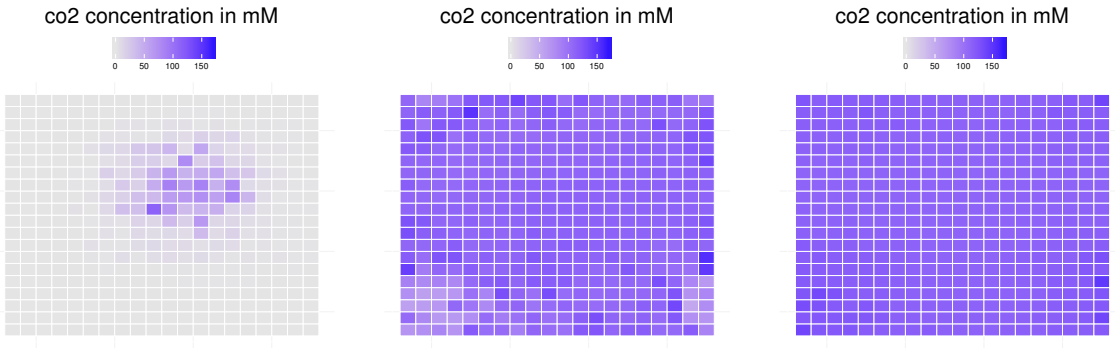
A



B



C



D

Figure 8: Population dynamics of the big *E. coli* model on a 20×20 grid, with bacterial movement (A) and concentrations of glucose (B), acetate (C) and CO_2 (D) (of time step 15, 40 and 50). The seed of the random number generator was set to 911.

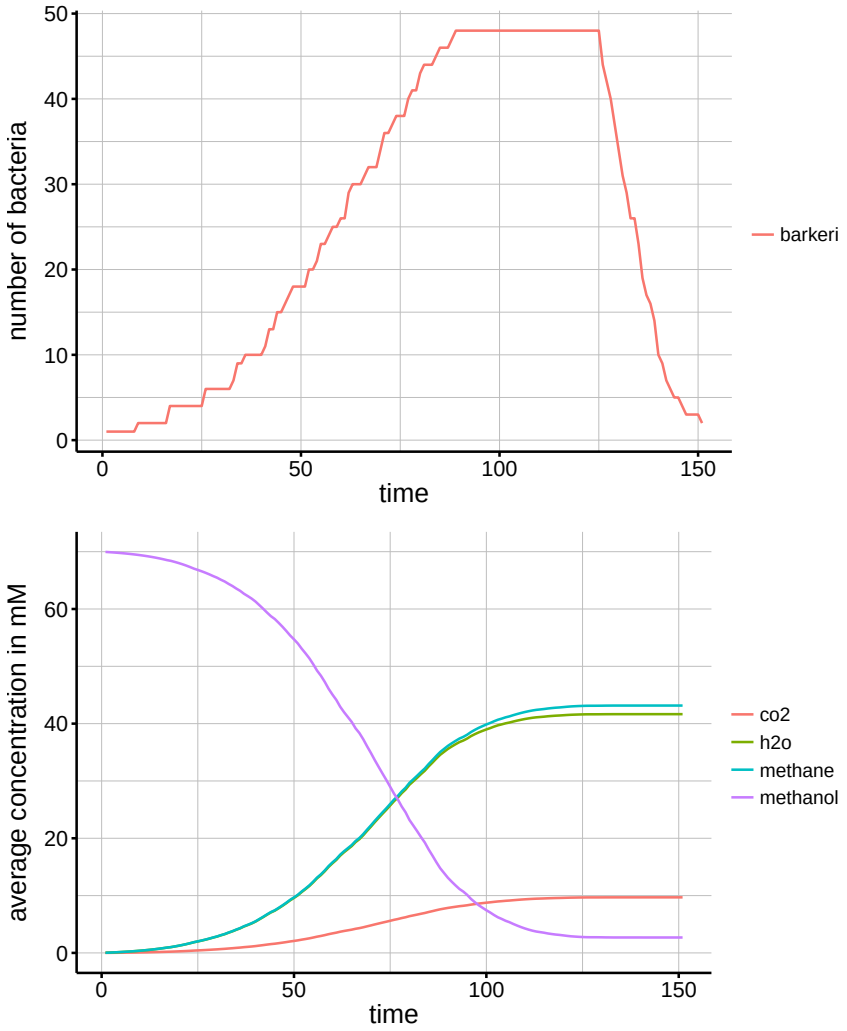


Figure 9: Population dynamics of the *M. barkeri* model on a 20×20 grid, with bacterial growth (A) and consumption/production of various metabolites (B). An initial concentration of 70 mmol per grid cell of methanol was added to the environment. The seed of the random number generator was set to 9659.

3.3 Mixed communities

3.3.1 Escherichia coli & Methanosarcina barkeri

The joint simulation of the big *E. coli* and *M. barkeri* was subjected to initial concentrations of glucose as a substrate for *E. coli* and methanol as a substrate for *M. barkeri*. In the first time steps the respective substrates were consumed and CO_2, acetate, ethanol, formate, water and methane were produced (Figure 13). Acetate was produced as a fermentation product of *E. coli* and consumed by *M. barkeri*. CO_2 was produced by *E. coli* and *M. barkeri*. *E. coli* reached the stationary phase in iteration 90. Since grid space was additionally occupied by *M. barkeri*, the *E. coli* population slightly grew after stationary phase of *M. barkeri* in iteration 125. Both populations died out after iteration 175. The doubling time in the exponential phase was estimated as 17 iterations (h) for *E. coli* and 12 iterations (h) for *M. barkeri*.

According the dispersion of the microbes on the grid environment, substrates were consumed and metabolites produced (Figure 16). The production of methane was found in grid positions occupied by *M. barkeri* and glucose consumption was found in position occupied by *E. coli*. In the later stages of growth acetate was consumed by *M. barkeri*.

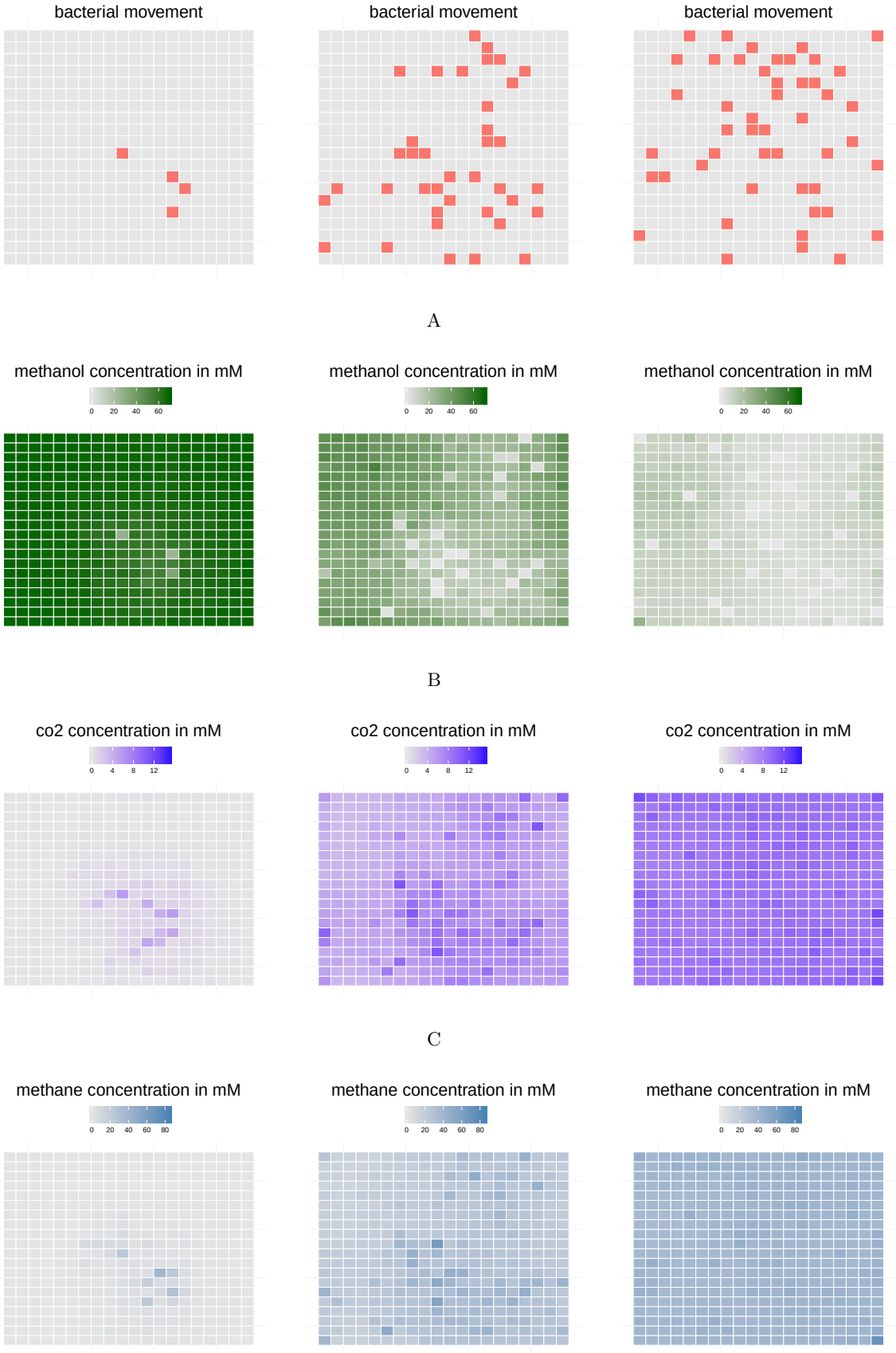


Figure 10: Population dynamics of the *M. barkeri* model on a 20×20 grid, with bacterial movement (A) and concentrations of methanol (B), CO_2 (C) and methane (D) (of time step 25, 75 and 100). The seed of the random number generator was set to 9659.

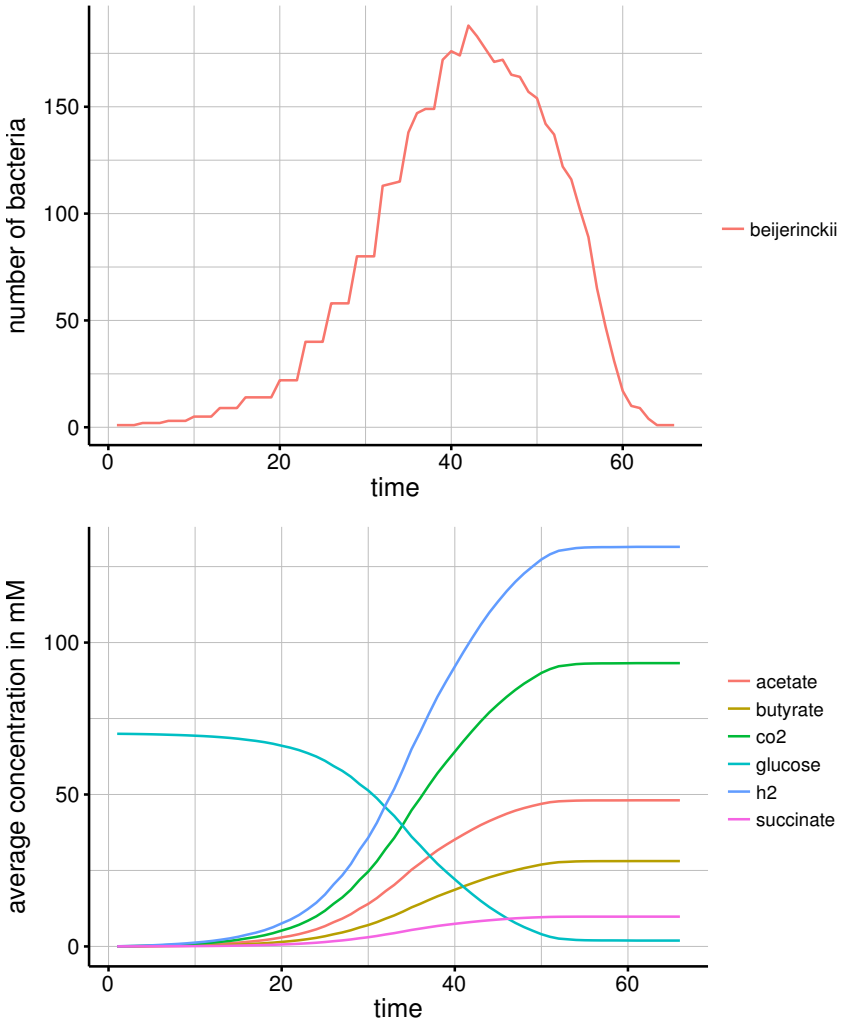


Figure 11: Population dynamics of the *C. beijerinckii* model on a 20×20 grid, with bacterial growth (A) and consumption/production of various metabolites (B). An initial concentration of 70 mmol per grid cell of glucose was added to the environment. The seed of the random number generator was set to 943.

3.3.2 Escherichia coli big & Clostridium beijerinckii

The joint simulation of the *E. coli* and *C. beijerinckii* was subjected to initial concentrations of glucose as a substrate for both organisms. In the first time steps the substrate was competitively consumed by both species and CO_2, acetate, ethanol, butyrate, succinate and hydrogen was produced (Figure 15). In comparison to the other metabolites hydrogen and CO_2 had the highest concentrations at the end of the simulation. Hydrogen, CO_2, butyrate and succinate were exclusively produced by *C. beijerinckii*, whereas ethanol was produced by *E. coli*. Acetate was produced by both species. *C. beijerinckii* reached the stationary phase at 55 iterations and *E. coli* at 60 iterations. Both organisms died at iteration 80. In the exponential phase *C. beijerinckii* had an approximate duplication time of 8 iterations and *E. coli* of 11 iterations.

According the dispersion of the microbes on the grid environment, substrates were consumed and metabolites produced (Figure 16). Hydrogen production was found in *C. beijerinckii* occupied grid positions and ethanol production in *E. coli* occupied positions.

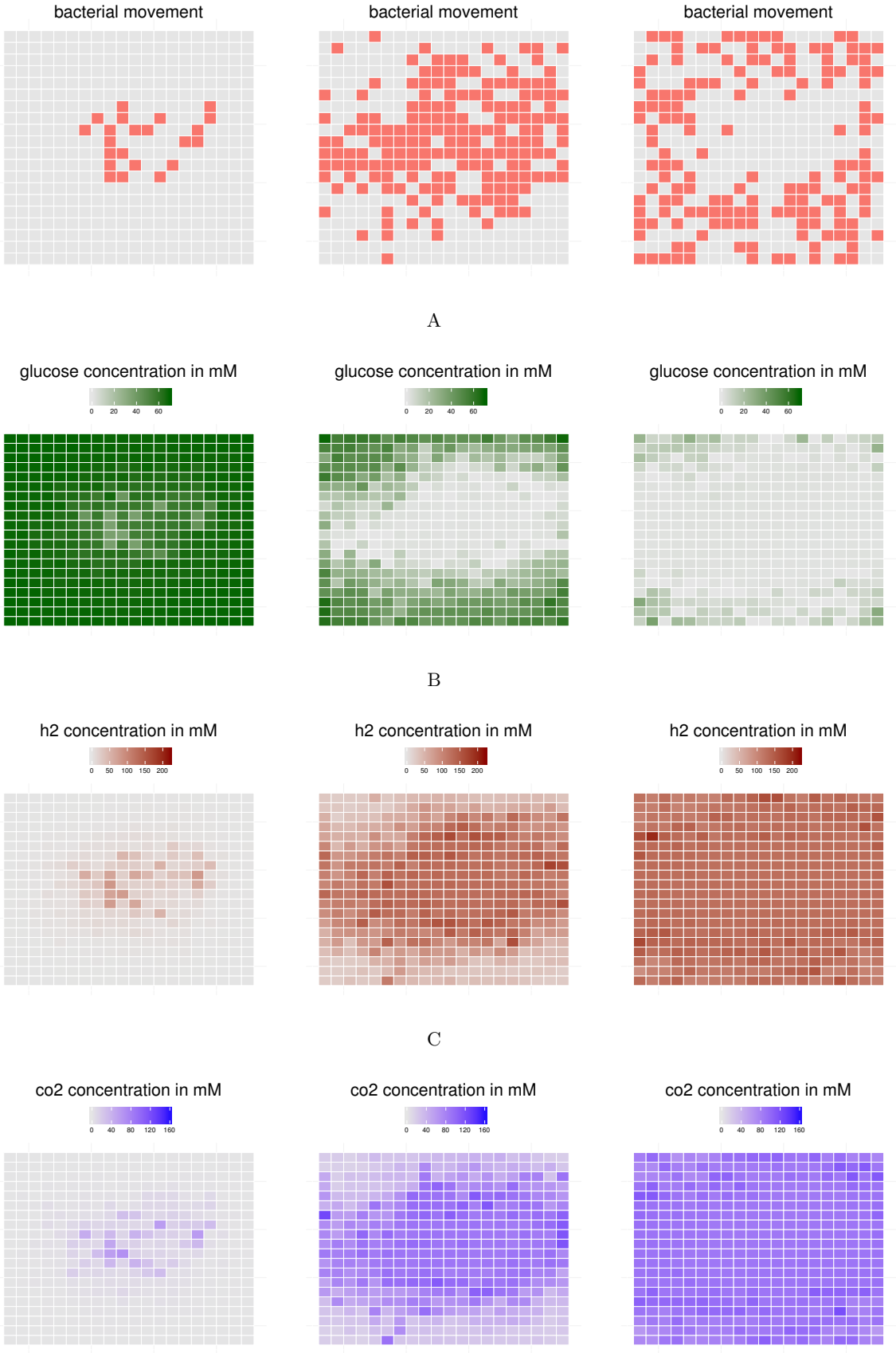


Figure 12: Population dynamics of the *C. beijerinckii* model on a 20×20 grid, with bacterial movement (A) and concentrations of glucose (B), hydrogen (C) and CO_2 (D) (of time step 20, 40 and 50). The seed of the random number generator was set to 943.

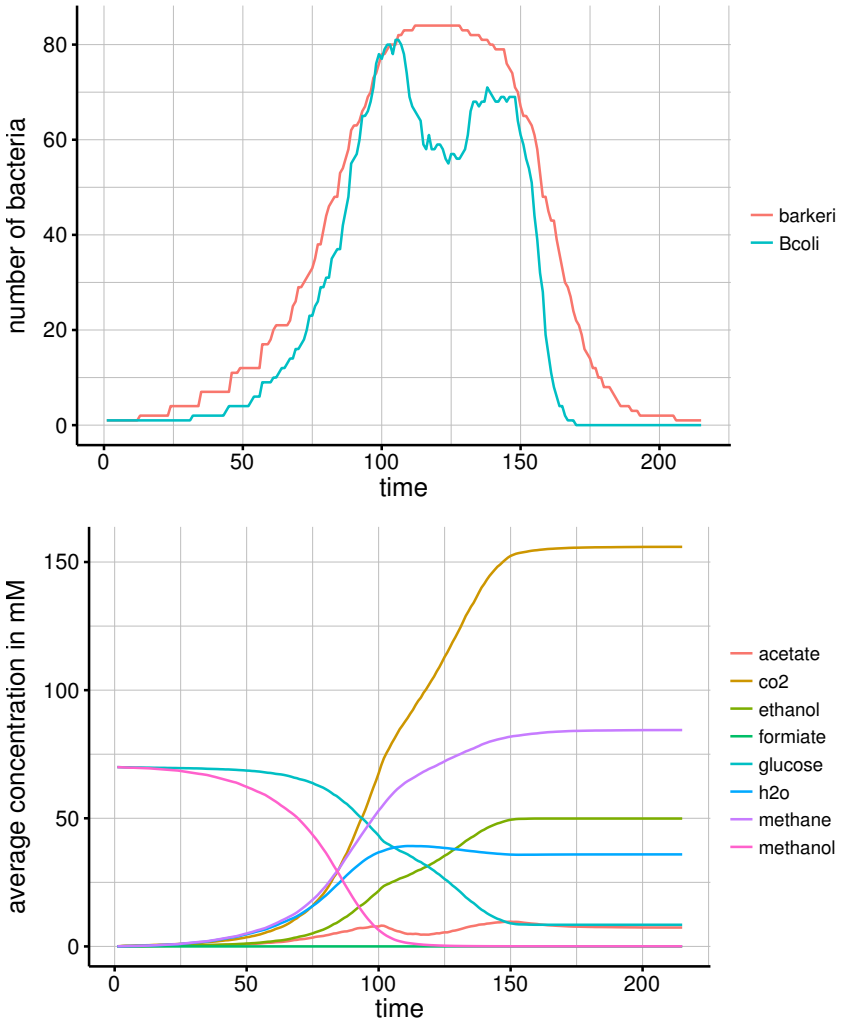


Figure 13: Population dynamics of the joint big *E. coli* and *M. barkeri* model on a 20×20 grid, with bacterial growth (A) and consumption/production of various metabolites (B). An initial concentration of 70 mmol per grid cell of glucose and methanol was added to the environment. The seed of the random number generator was set to 4612.

3.3.3 Clostridium beijerinckii & Methanosarcina barkeri

The joint simulation of the *M. barkeri* and *C. beijerinckii* was subjected to initial concentrations of glucose as a exclusive substrate for *C. beijerinckii*. In the first time steps the substrate was consumed by *C. beijerinckii* and CO_2, hydrogen, acetate, butyrate and succinate was produced (Figure 17). In comparison to the other metabolites hydrogen and CO_2 had the highest concentrations at the stationary phase of *C. beijerinckii* reached at approximately 60 iterations. The population of *C. beijerinckii* died out at iteration 80 and *M. barkeri* started growing exponentially by taking the produced hydrogen and CO_2 as substrates. In the later phases of growth the produced acetate was used as a substrate. The stationary phase of *M. barkeri* was reached at iteration 170. During growth *M. barkeri* produced methane and water by consuming CO_2 and almost the total hydrogen produced by *C. beijerinckii*. In the exponential phase *C. beijerinckii* had an approximate duplication time of 8 iterations and *M. barkeri* of 12 iterations.

According the dispersion of the microbes on the grid environment, substrates were consumed and metabolites produced (Figure 18). Hydrogen production was found in *C. beijerinckii* occupied grid positions and methane production in *M. barkeri* occupied positions.

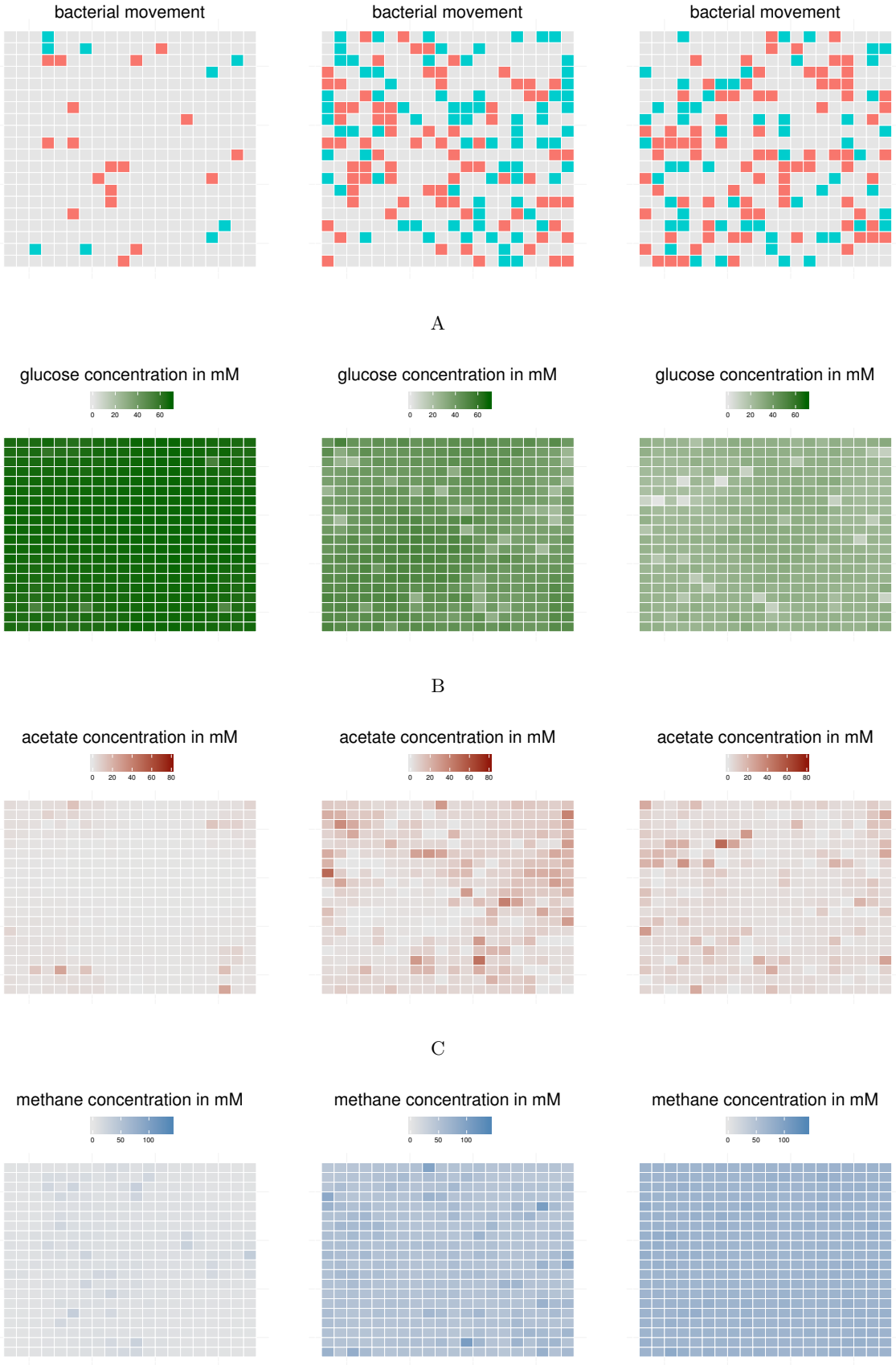


Figure 14: Population dynamics of the joint big *E. coli* and *M. barkeri* model on a 20×20 grid, with bacterial movement (A) and concentrations of glucose (B), acetate (C) and methane (D) (of time step 60, 100 and 125). The seed of the random number generator was set to 4612.

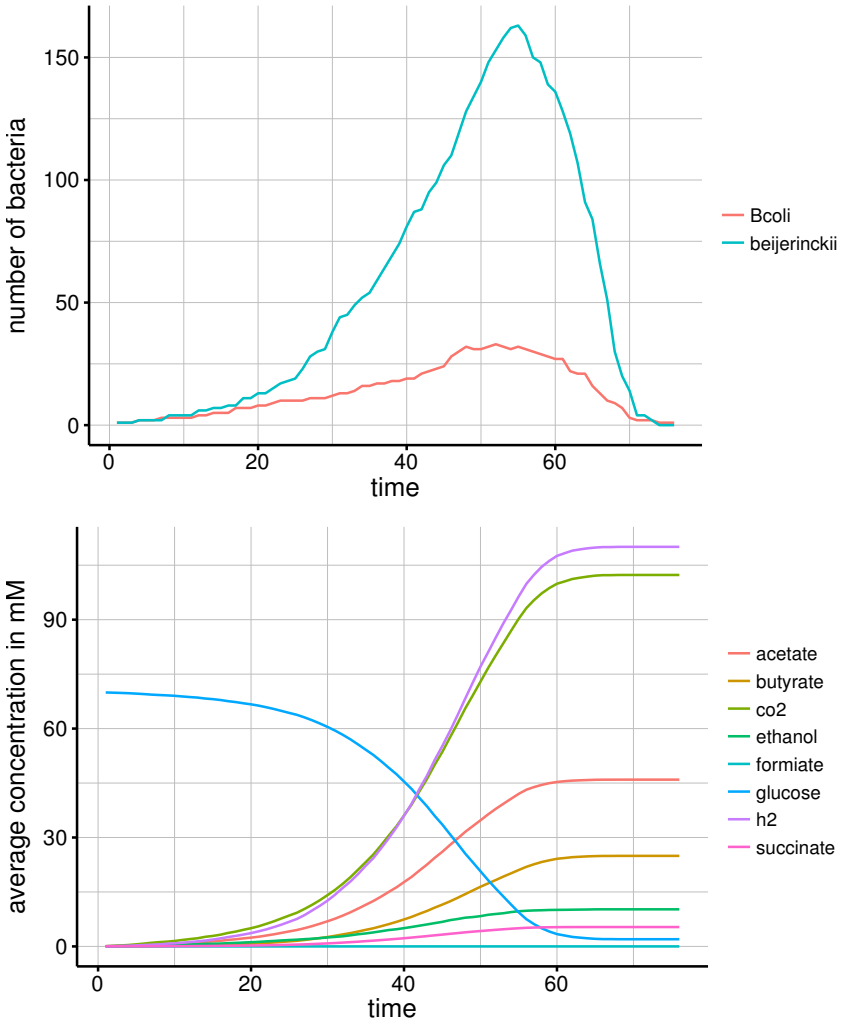


Figure 15: Population dynamics of the joint big *E. coli* and *C. beijerinckii* model on a 20×20 grid, with bacterial growth (A) and consumption/production of various metabolites (B). An initial concentration of 70 mmol per grid cell of glucose was added to the environment. The seed of the random number generator was set to 5147.

4 Discussion

4.1 General discussion

4.1.1 Diffusion and movement

In agent based modeling two different modes for updating can be distinguished: A synchronous mode updates all cells simultaneously, i.e. local changes are stored in a temporary copy and will be updated after the computation of all cells. Contrary to this, a asynchronous mode updates changes immediately ([**Matthies2002**] p. 92).

In our study we implemented for the diffusion rule applied to agents a naive model, which relies on the asynchronous update with randomly chosen cells. For the diffusion of substrates this method is preferred, because i) synchronous updates would violate the conservation laws by the production of additional metabolite concentrations and ii) non-random asynchronous updates cause a biased diffusion direction [**Bandman1999**]. As indicated in Figure 3 the spreading of metabolite concentrations causes the increase of entropy in the system, which is also observed as a physical phenomenon in microbial communities [**Wetzel93**]. Additional refinements can be realized with more sofisticated diffusion models such as block-rotation [**Bandman1999**] or the discrete diffusion model by Grajdeanu

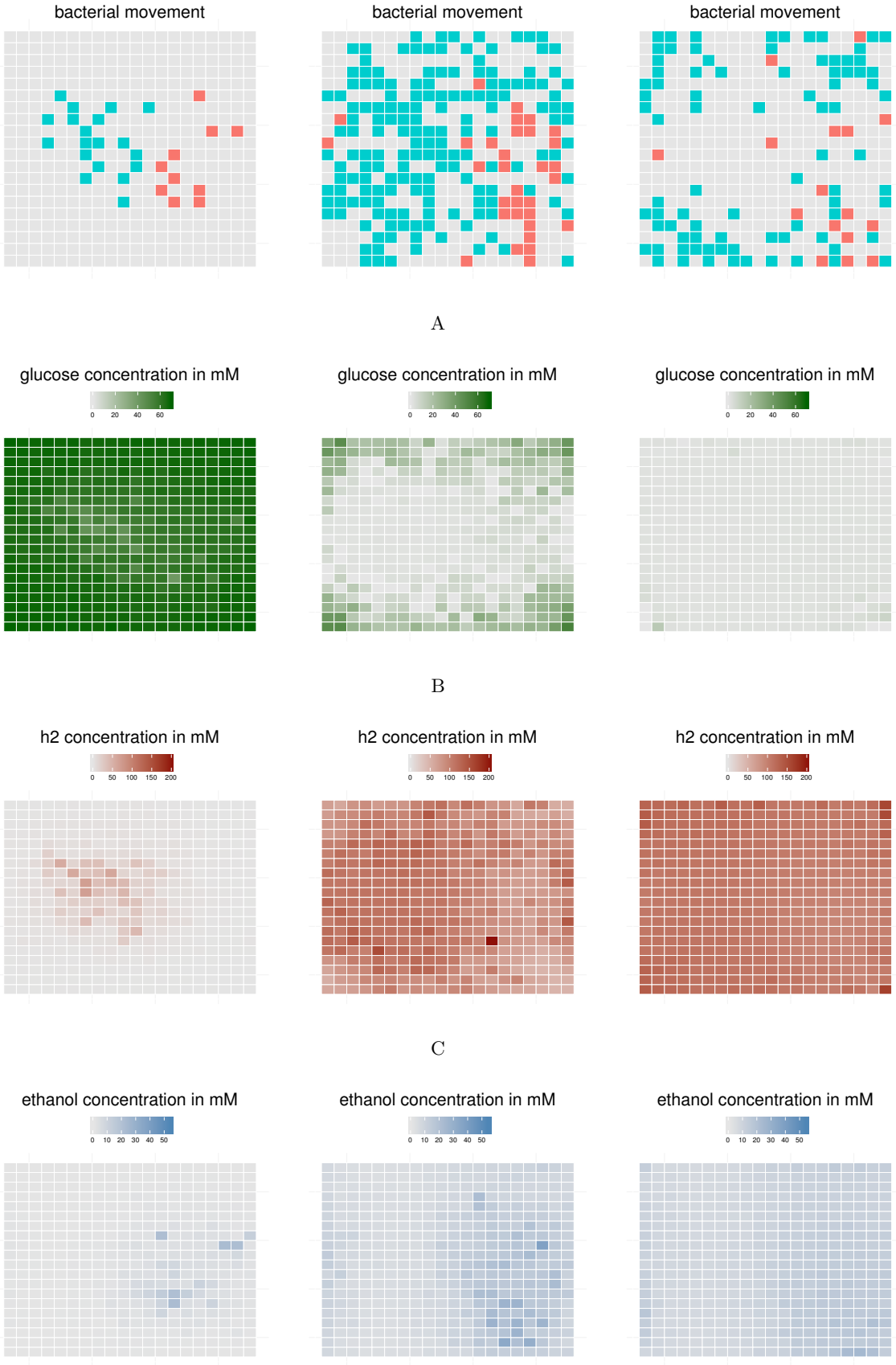


Figure 16: Dynamics of the joint big *E. coli* and *C. beijerinckii* model on a 20×20 grid, with bacterial movement (A) and concentrations of glucose (B), hydrogen (C) and ethanol (D) (of time step 25, 55 and 65). The seed of the random number generator was set to 5147.

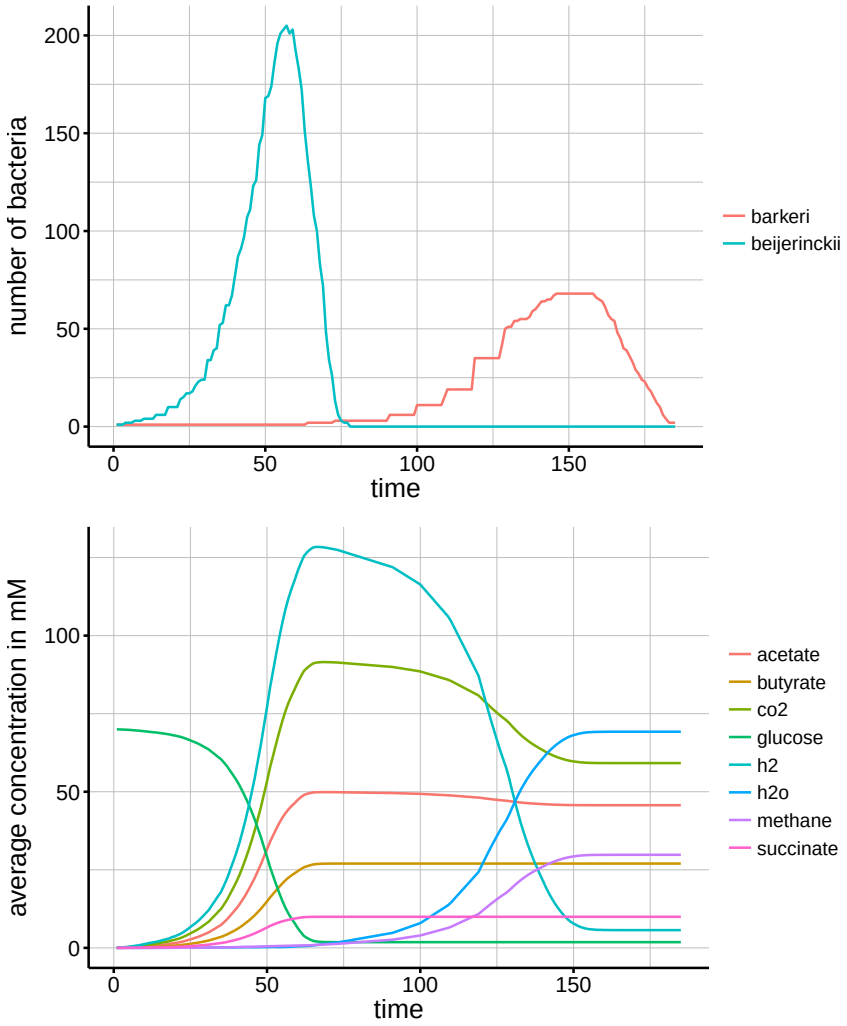


Figure 17: Population dynamics of the joint *M. barkeri* and *C. beijerinckii* model on a 20×20 grid, with bacterial growth (A) and consumption/production of various metabolites (B). An initial concentration of 70 mmol per grid cell of glucose was added to the environment. The seed of the random number generator was set to 2942.

[**Grajdeanu2007**]. Different diffusion coefficients of certain metabolites could also be included to model the varying dispersal speeds.
The bacterial movement was implemented similar to the diffusion model as a random spread on the grid environment (Figure 2). Since most bacteria are able to sense the metabolite concentrations in their environment and show a directed motion to the substrate of choice [**Francisco13**], the movement model can be refined by interaction of the bacterial agents with the substrates. A chemotaxis model would also probably allow the observation of more complex behaviours such as the aggregation to certain parts of the grid environment.

4.1.2 Time consumption

FBA calculation of substrate exchange can take 2 *min* per time step for 180 bacteria of the big *E. coli* model (1972×2382 metabolites and substrates). This leads to an overall calculation time of several hours for the *Bcoli* population model. In comparison to this, the *E. coli* (77×77) core model needed only 2 s per time step for 180 bacteria. Almost all of the required time is spent for fba calculation. For this reason further improvements can be done by:

1. Hashing: Implementation of a fba memory table, which saves prior calculations.

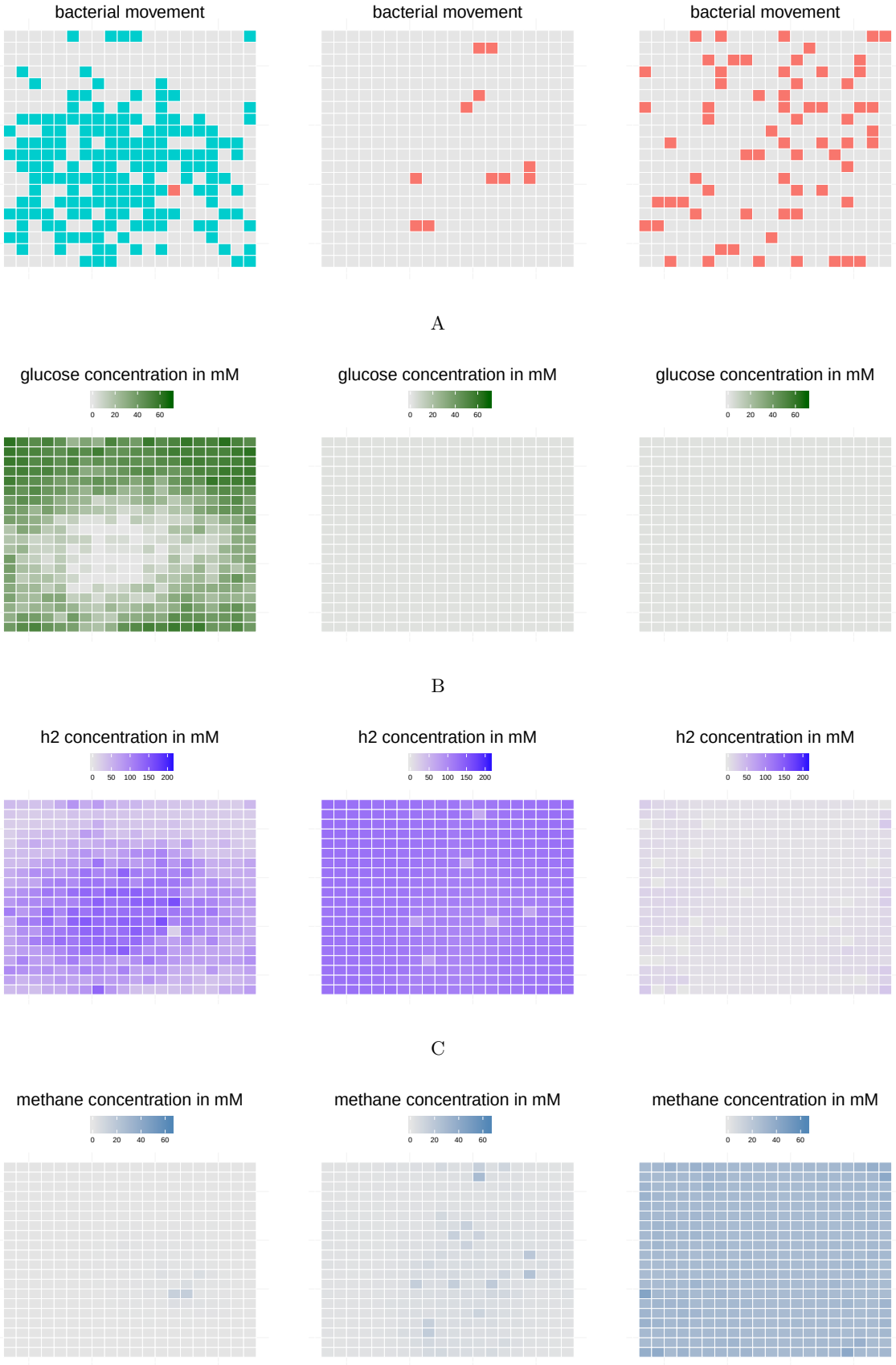


Figure 18: Dynamics of the joint *M. barkeri* and *C. beijerinckii* model on a 20×20 grid, with bacterial movement (A) and concentrations of glucose (B), hydrogen (C) and methanol (D) (time step 50, 100 and 150). The seed of the random number generator was set to 2942.

2. Starting base: *lpsolve* offers since version 5.1.05 the possibility to find a basis according to some guessed vector (old solution). With this basis the optimal solution of the optimization problem might be found faster [**warmstart**].

We implemented already a first form of hashing, however the speed improvements were so far not quantified and compared.

4.2 Population models of single organisms

All population models show similarities in their growth curves (Figure 5, 7, 9 and 11). Roughly, the observed growth can be separated into three phases: the exponential, the stationary and the death phase. Those phases are also observed in experimental studies of microbial growth [**Varma1994**]. During the exponential phase all substrates are efficiently used by the fba to accumulate biomass, which is then used for duplication. In the stationary phase almost all substrates were exploited and the fba does not find any feasible solution, which results in the reduction of biomass. Subsequently, bacterial agents are removed, if the biomass is below zero, leading to the death phase.
According to experimental studies [**Varma1994**], the time for *E. coli* to reach the stationary phase is about 8 h, which is in contradiction to the observed time of approximately 39 h in our study (Figure 5). This can be explained by the used artificial media composition, which might not be sufficient for an optimal growth. Moreover, the substrate concentrations were set to not empirically justified arbitrary values. Further refinements of substrate concentrations could thus increase the accuracy of the model to match experimental results. Additionally, the constraints of the used exchange reactions can also be refined to more realistic values.
The observed growth of *E. coli* (Figure 5) can be explained mainly by the aerobic respiration of glucose:

$$C_6H_{12}O_6 + 6O_2 \rightarrow 6CO_2 + 6H_2O$$

where one molecule of glucose and 6 molecules of oxygen are consumed to produce 6 molecules of CO_2 and water. In accordance to the stochiometry of this reaction more oxygen was consumed by the model compared to glucose (Figure 7). Additionally, various fermentation products are produced via mixed acid fermentation under aerobic conditions (Figure 5), which is in concordance to experimental studies [**Sunya13**]. In these studies, formate is only produced under anaerobic conditions. Interestingly, the production of formate is observed under aerobic conditions in *E. coli* core, whereas *E. coli* big does not produce formate (Figure 5). Furthermore, the smaller *E. coli* core model produces overall more fermentation products relative to CO_2. This can be explained by the additional pathways present in the more representative larger *E. coli* model, which make the complete mineralization of glucose to CO_2 more preferable in the fba optimization. Since the complete mineralization of glucose would be even more preferable in the larger *E. coli* model, the applied uptake constraints (Table 1) were set to allow the production of fermentation products.
The growth of *M. barkeri* (Figure 9) can be explained by the fermentation of methanol to methane with

$$4CH_3OH \rightarrow 3CH_4 + CO_2 + 2H_2O$$

where 4 molecules of methanol are consumed to produce 3 molecules of methane, 2 of water and 1 molecule CO_2. The stochiometry of this reaction was fulfilled by the higher production of methane compared to CO_2 (Figure 9). However, water was, similar to methane, produced in high amounts, which can be explained by additional reactions in the model, which might lead to water production.

In experimental studies the stationary phase of *M. barkeri* under the consumption of methanol is reached after 72 h [**Hippe79**] which is in concordance to our observed time of approximately 80 h.
C. beijerinckii is able to convert glucose to various fermentation products [**Ezeji03**] via butyrate fermentation. To remove reducing equivalents hydrogen is formed, which is also observed in our model (Figure 11). The time for *C. beijerinckii* to reach the stationary phase with glucose as a substrate was experimentally determined to approximately 60 h [**Ezeji03**], which is in contradiction to the observed time of 40 h (Figure 11). To match the experimental results the constraints of the exchange reaction can be refined.
Growth deviations (especially doubling time) have to be checked with multiple runs and a wide range of circumstances to guarantee more representative results.

4.3 Interactions in mixed communities

Since *M. barkeri* can also utilize hydrogen and CO_2 for methane production and *C. beijerinckii* is able to produce those metabolites, both organisms are good candidates to study inter-species hydrogen transfer and syntrophy. Furthermore, co-culture studies on *M. barkeri* with hydrogen producing microbes [**Winter79**] have demonstrated the capabilities of *M. barkeri* to interact with other organisms.
In our mixed community model (Figure 17) we could observe the usage of *C. beijerinckii*'s produced hydrogen and CO_2 by *M. barkeri*. However, in contrast to experimental observations showing the parallel growth of both organisms [**Winter79**], we observed a delayed growth of *M. barkeri* after the death of *C. beijerinckii* (Figure 17). This can be explained by the limited space on the grid environment, which favoured the initial growth of *C. beijerinckii*, since enough substrates were available. Consequently, *M. barkeri* did not have enough space to duplicate and could only spread after the other organism died out (Figure 18). Nevertheless, *M. barkeri* was still able to survive, due to the hydrogen and CO_2 support by *C. beijerinckii*. With the production of *M. barkeri*'s essential metabolites, *C. beijerinckii* thus shaped the environment of the other organism.
In the co-culture of *C. beijerinckii* with *E. coli* we observed competitive interactions, where both organisms competed for the same substrate. In this context, *C. beijerinckii* was able to outcompete the other bacterium with an overall higher growth (Figure 16). *E. coli* was not able to show high growth rates, since the main substrate was consumed by the competitor.
These results might indicate a more efficient anaerobic degradation by *C. beijerinckii* compared to *E. coli*. Another explanation can be the pure chance of *C. beijerinckii* having better starting conditions, which lead to a higher growth (positive feedback). Further experimental co-culture studies might validate this observation.
In the co-culture of *M. barkeri* with *E. coli* we observed space competitive effects, in which the space was not sufficient to harbour both organisms at their stationary phase (Figure 14). The growth curves of both species were similar to each other, which speaks for no competitive metabolic interactions as observed in the *C. beijerinckii*, *E. coli* joint model (Figure 15). Moreover, *M. barkeri* was able to utilize acetate produced by *E. coli* (Figure 13).

4.4 Conclusions & outlook

In the last years a lot of work has been done to build up models for single organisms [**lewis2012**]. This models are capable to reproduce experimental results and permit even quantitative phenotypic predictions [**mccloskey**]. Recently, voices were being raised to point out the future direction of research on community systems biology:

> ,,We anticipate that, through the use of bottom-up approaches supplemented with meta-omics data, the success of systems biology for individual organisms will now be extended to communities of organisms, and in particular to microbial communities." [**cosys**]

We implemented such a bottom up approach in our agent based model, which uses the established strength of individual models to simulate metabolic interactions and even syntrophy between communities of *in silico* species.