iDynoMiCS: Next-generation Individual-based Modelling of Biofilms

Supporting Information

Laurent A. Lardon*[,1,2], Brian V. Merkey*[,1,3], Sónia Martins[4], Andreas Dötsch[5], Cristian Picioreanu[6], Jan-Ulrich Kreft[4], Barth F. Smets[a,1]

[1]Department of Environmental Engineering, Technical University of Denmark, Bygningstorvet 115, 2800 Kgs. Lyngby, Denmark
[2]Laboratory of Environmental Biotechnology, INRA, Avenue des étangs, 11100 Narbonne, France
[3]Department of Engineering Sciences and Applied Mathematics, Northwestern University, 2145 Sheridan Rd., Evanston, IL 60201, USA
[4]Centre for Systems Biology, School of Biosciences, The University of Birmingham, Edgbaston, Birmingham B15 2TT, UK
[5]Chronic Pseudomonas Infections Group, Helmholtz Centre for Infection Research, Inhoffenstrasse 7, D-38124 Braunschweig, Germany
[6]Department of Biotechnology, Delft University of Technology, Julianalaan 67, 2628 BC Delft, The Netherlands
*These authors contributed equally to this work
[a]Corresponding Author: bfsm@env.dtu.dk

I Introduction

The Supplementary Materials provides further details of the iDynoMiCS model description and validation against previous models.

II Random Number Generation

Random numbers are generated using the Mersenne Twister MT19937 algorithm (Makoto and Takuji 1998) because of its satisfactory statistical properties and because its very long period before numbers will repeat ($2^{19937} - 1$) ensures that simulations will not exhaust the stream of unique random numbers. iDynoMiCS will output the state of the random number generator upon completion of a simulation, and if a particular simulation is to be repeated with identical parameters and initial conditions a given number of times the random number generator is not restarted but instead reads the previous state from file in order to use non-overlapping sets of pseudo-random numbers. This enables replication of runs to study the effect of stochasticity. Random numbers drawn from a normal

distribution are restricted to be in the range mean ± 2 SD. Moreover, the randomly varied parameter is guaranteed to have the same sign as the default value of that parameter.

III Boundary Conditions

Here are more complete descriptions of the boundary conditions included in the initial iDynoMiCS release.

- ***No-Flux Boundary***: This boundary is impermeable to solutes and agents; as a consequence, the normal components of solute concentration gradients will be zero at this boundary. Mathematically, the boundary condition for the solutes is given as:

$$\nabla S(\vec{x}) \cdot \hat{n}\big|_{\Gamma} = 0\,, \tag{III-1}$$

  where $S(\vec{x})$ is the solute concentration in the domain, $\hat{n}$ is the boundary normal vector and Γ is the boundary of interest. Agents attempting to cross the boundary are prevented from doing so (Figure S1).
- ***Constant Concentration Boundary***: The boundary represents, for example, the connection to a larger system where the concentration can be considered constant. The solute concentrations at this boundary are fixed, and agents crossing this boundary are considered to have entered the planktonic bulk domain. The solute condition is given mathematically by:

$$S(\vec{x})\big|_{\Gamma} = S_B, \tag{III-2}$$

  where S is the solute concentration in the domain, S_B is the concentration at the boundary, and Γ is the boundary of interest.
- ***Variable Concentration Boundary***: This boundary simulates the connection to a larger bulk liquid subject to a dilution process, as would be typical for a reactor. Behaviour for agents does not differ from the constant concentration boundary case (and so within the computational domain the boundary condition is identical), but solute dynamics in the bulk compartment require an additional computational step. In this step, ordinary differential equations describing the reactions occurring in the biofilm and the hydraulic processes affecting the bulk liquid are solved in order to determine the bulk concentration at the next time-step.
- ***Membrane Boundary***: A membrane boundary has a selective permeability, meaning it behaves like a zero-flux boundary for agents and most of the solutes, but for selected solutes includes specification of the diffusivity in the membrane and the opposing-side solute concentration. Then the boundary condition is given as:

$$D_S \nabla S(\vec{x}) \cdot \hat{n}\big|_{\Gamma in} = D_M \nabla S(\vec{x}) \cdot \hat{n}\big|_{\Gamma out}\,, \tag{III-3}$$

  where D_S is the solute diffusivity inside the computation domain, D_M is the solute diffusivity inside the membrane, and the boundary designations Γ_{in} and Γ_{out} represent derivatives taken on the inside and outside of the boundary, respectively.

- ***Periodic Boundary***: It is computationally unfeasible to simulate a micro-scale world on a macro-scale level, so oftentimes a small spatial sub-region is assumed to represent the system as a whole; in this case, periodic boundaries are used to remove artificial edge effects by assuming the simulated region adjoins other, similar, regions. As a consequence, boundaries in some chosen directions (generally for movements parallel to the substratum) are periodic, which means that the solute concentrations and solute gradients are constant across the boundary, and that agents travelling through one boundary will be translated to the other side of the domain (Figure S1). For the solutes, this condition is expressed as:

$$S(\vec{x})\big|_{\Gamma near} = S(\vec{x})\big|_{\Gamma far} \qquad \text{(III-4)}$$
$$\nabla S(\vec{x})\big|_{\Gamma near} = \nabla S(\vec{x})\big|_{\Gamma far},$$

  where Γ_{near} and Γ_{far} represent the two boundaries connected periodically.

IV Defining the Diffusion Layer

The diffusion layer is a region separating the biofilm from the bulk compartment, and in this region the solute concentrations vary spatially only due to diffusion, in comparison to the biofilm region where concentrations vary due to diffusion and to production/consumption, or the bulk compartment region where the solute concentrations do not vary spatially. The diffusion layer thickness l_{BL} is specified by the user, and is generally on the order of tens of microns in thickness. During a simulation, the location of the boundary between the diffusion layer and bulk compartment (Γ_{BL} in Figure 1 in the manuscript) must be calculated at each step to account for any change in the position of biofilm surface during that step. The location of the boundary is found by a dilation process, where for each point above the biofilm surface a spherical region of radius l_{BL} is scanned for the presence of biomass or carrier. If the surrounding sphere contains biomass or carrier, then the point is considered to be within the diffusion layer, but if the sphere does not contain biomass or carrier then the point is considered outside the diffusion layer.

V Computing the Pressure Field

The positions of agents within the biofilm are affected on a small scale by local mechanical forces, such as shoving by neighbours, and also by larger-scale movements of biofilm biomass driven by the pressure generated by an increase (growth) or decrease (decay) of biovolume. A novel feature of iDynoMiCS is that the particles (agents) are moved by both small and large scale mechanical forces. As a supplement to the description of the pressure field in the main text,

Figure S2 shows the distribution of pressure in a biofilm growing under diffusion-limited conditions. The difference in growth rates in each region results in a distribution of

pressure: maintenance terms create a shrinking of the biofilm toward the bottom (negative pressure, blue and cyan), while high substrate concentrations yield biofilm growth at the top (positive pressure, yellow and orange). A zero-pressure line (green) separates these two regions.

VI Agent Shoving

During each global time-step, agent divisions (Figure S3) and agent growth will lead to many cases where neighbouring agents overlap. A relaxation algorithm is used to find iteratively the new overlap-minimizing steady-state configuration of agent locations at the end of each time-step (Algorithm S1). During this process, any overlap of agents is estimated not on the basis of the agent radius $\varphi_{j,Total}$ (the total radius includes all interior compartments as well as any capsule), but rather on the basis of a shoving radius $\varphi_{j,Shove}$, where $\varphi_{j,Shove} = k_{Shove} \cdot \varphi_{j,Total}$ (Figure S4). The shoving factor k_{Shove} allows adjustment to the degree of packing of the agents, and the shoving radius is used to calculate the overlap δ between two adjacent agents:

$$\delta = d_{1,2} - k_{Shove} \cdot (\varphi_{1,Total} + \varphi_{2,Total}). \quad \text{(VI-1)}$$

In this expression, $d_{1,2}$ is the distance between the centres of the two agents. Within each iteration step, each agent is visited, and for each neighbour of the currently focal agent any overlap is addressed with equally far but opposite movements of the agents (Algorithm S1). However, rather than moving the agents immediately, all such overlap-relieving movements are summed and the net movement is applied after all such movements have been calculated (Algorithm S1). Saving the movements and applying them only at the end eliminates any bias that may arise when some agents are moved before others.

VII Erosion and Detachment

iDynoMiCS implements a choice of empirical detachment functions, where the rate of detachment is typically a function of biofilm height. Numerically, this is solved using standard level-set methods. More formally, if $\vec{x}$ is the coordinate vector of a point located on the biofilm-liquid interface, and $\hat{n}(\vec{x})$ a unit vector normal to the surface at that point $\vec{x}$, then the front velocity due to erosion forces is given by:

$$\frac{d\vec{x}}{dt} = -F_{Det}(\vec{x}) \cdot \hat{n}(\vec{x}), \quad \text{(VII-1)}$$

where $F_{Det}(\vec{x})$ is a detachment speed function (units $(\mu m / hr)$) that may in principle take any form (see below for common forms). This detachment speed can be propagated to all points in the domain using the Fast Marching Method (Sethian 1999), a numerical technique used to calculate the time-of-crossing for a moving interface. The method is used to compute at any point in the biofilm the time required for the interface to reach this point, assuming the interface velocity does not vary in time (which for a single time-

step it does not). This is accomplished by starting from the current interface and computing the time-of-crossing for points near the interface, and then proceeding towards the biofilm interior to extend the time-of-crossing to neighbouring points. By this iterative process it is possible to define the whole set of surfaces describing the time course of the boundary. Figure S5 shows an example of the advance of erosion by contour lines, with each line corresponding to the position of the erosion front at different points in time under the assumption of zero growth.

Erosion effects are applied during each time-step by first computing the time-of-crossing map for the current biomass distribution. The time-of-crossing value T_i is calculated for each grid element *i*, and T_i specifies the time at which the biofilm interface would cross that point *i* if the biofilm/liquid interface eroded at a constant speed. The values of T_i will tend to be larger for a slower-moving interface (low erosion), and will be smaller for a faster-moving interface (high erosion). After being computed, the time-of-crossing value T_i for each grid element is compared to the global timestep Δt_{global}, yielding a ratio *r*:

$$r \equiv \frac{\Delta t_{global}}{T_i}. \tag{VII-2}$$

This ratio *r* captures the fraction of mass in an element that should be lost through erosion during a single timestep: during erosion, the mass of all agents whose centres lie in that element is decreased by *r*%, and an agent is removed if its mass becomes zero. Note that *r* is capped at a value 1 if the time-of-crossing is smaller than the global time-step. For high erosion speeds the time-of-crossing value T_i will be smaller, and so a larger fraction of the biomass will be removed. The time-of-crossing map is recomputed during each step in order to account for any structural changes that may have occurred during the current step.

The detachment speed $F_{Det}(\vec{x})$ may in principle take any form, but iDynoMiCS includes two common detachment speed functions. The first is a height-based speed:

$$F_{Det}(\vec{x}) = k_{Det} \cdot h^2, \tag{VII-3}$$

where k_{Det} is the detachment strength coefficient (with units of $(\mu m \cdot hr)^{-1}$), and h is the local height above the carrier. This speed function was first introduced for 1D continuum models, and is preferred because it leads to a steady-state biofilm of constant thickness (Wanner and Gujer 1986). The second detachment speed included in iDynoMiCS is a height- and biomass-based detachment:

$$F_{Det}(\vec{x}) = k_{Det} \frac{h^2}{\rho_X}, \tag{VII-4}$$

where k_{Det} is again the detachment strength coefficient (with units of $fg/(\mu m^4 \cdot hr)$), h is again the local height above the carrier, and ρ_X is the local biomass concentration in the biofilm. This detachment function will lower the detachment speed when the local biomass concentration is high (i.e. denser parts of the biofilm are more difficult to remove).

In addition to erosion on the biofilm surface, iDynoMiCS can also make use of an imposed maximum thickness parameter that may be used to model a constant-depth film fermenter (Peters and Wimpenny 1988) or to contain the biofilm within the computational domain. When such a thickness is imposed, any agent crossing this boundary is removed from the biofilm.

VIII Individual-based Chemostat

The chemostat version is mainly used as a reference case without spatial structure to evaluate the effect of spatial structure so characteristic for biofilms. In the chemostat mode, some of the steps in each iteration of the algorithm are skipped for efficiency, namely the stages of spatial positioning of agents, computation of the pressure field and pressure-driven movements, and shoving and biomass detachment. In a chemostat, the medium is assumed to be uniformly mixed, and hence all agents 'see' the same concentration for all solutes. The concentrations of the solutes are governed by the processes of dilution (inflow and outflow at the same rate) and bacterial growth. The dilution rate is used to calculate the fraction of agents to be removed stochastically from the system during the current time step. Because the chemostat equations are typically stiff ODEs, the diffusion-reaction problem is solved using a modified Rosenbrock pair formula based on partial derivatives (Shampine 1982; Shampine and Reichelt 1997). The method is very dependent on an accurate Jacobian matrix (which in our case is calculated analytically), and in this method the function is evaluated twice, yielding an intermediate solution estimate F_1, which is used to obtain the solution for the next time-step, F_2. If the calculated error is smaller than a given tolerance, then the step is considered successful and the predicted F_2 solution will be used as the initial state for the next step; otherwise, the solver time step is decreased according to the standard rule used in numerical integration for initial value problems (Gear 1971), and the step is carried out once more.

Note that for stochastic dilution in small systems (few cells; small volumes), any variation in the number of agents to be removed for each species can lead to discrepancies with the deterministic solution (as was the case in the chemostat model verification). Whether these differences will lead to divergent results will depend on the feedbacks in the system. In our test case, results converged to the deterministic steady state solution despite strong initial differences due to the stochastic initialization of the IbM. In a bigger system, these stochastic effects will be diminished and results converge toward the mass-action solution; therefore it is recommended to vary the system size to evaluate the importance of stochasticity.

IX Example Reaction Description

Table S1 illustrates a simple example reaction matrix describing growth, maintenance and decay reactions. It also illustrates how reactions can affect and be catalysed by different compartments of the structured individual. In this example, the entity is

composed of three compartments: the active biomass (C_X), an inert compartment (C_I) and a capsule of EPS (C_{EPS}); here EPS is considered to be a product of the growth reaction. It is assumed that all active reactions are independent and that the global variation of mass for a given solute or agent is obtained by summing the uptake rates of all active reactions. In this example we have the following for the biomass compartment:

$$\frac{dX_A}{dt} = r_A = \sum_{i \in X} Y_{X,i} \cdot r_i = (1 - Y_E) \cdot r_1 - r_2 - r_3 = ((1 - Y_E) \cdot \gamma_1 - \lambda_2 - \lambda_3) \cdot X_A$$

The expression for the EPS capsule takes a similar form:

$$\frac{dX_{EPS}}{dt} = r_{EPS} = \sum_{i \in EPS} Y_{EPS,i} \cdot r_i = Y_E \cdot r_1 - r_4 = Y_E \cdot \gamma_1 \cdot m_X - \gamma_4 \cdot X_{EPS} \quad \text{(IX-2)}$$

and the rate expressions for the solutes and other particulate components are constructed in the same manner.

X Validation Against Benchmark Problem BM3

The iDynoMiCS platform has been validated against the benchmark for multi-species biofilm models proposed by the International Water Association (IWA) task group on biofilm modelling in order to compare the variety of modelling approaches used (Noguera and Picioreanu 2004; Rittmann et al. 2004). The benchmark problem is meant to furnish a realistic two-species biofilm wastewater treatment scenario, where the complexity of the system is however constrained by the fact that all models need to be able to simulate this scenario. Therefore, the detachment follows an all-or-nothing behaviour: any part of the biofilm exceeding a given maximal thickness is removed. The benchmark includes an autotrophic bacterial species oxidising ammonia to nitrate and a heterotrophic species oxidising COD. Both species carry out growth, maintenance and decay reactions as described in Table S2. Parameters describing the benchmark problem are given in Table S3, and the environmental conditions in Table S4. Finally, Table S5 compares the output of the iDynoMiCS simulations (two replicate simulations gave the same output) with the numerical results (only steady state concentrations of the solutes in the bulk, but other state variables are coupled) of four different previous models, published in (Noguera and Picioreanu 2004). Since the solute concentrations are not independent variables, we have used Hotelling's T^2 test, the multivariate extension of the t-test (Anderson 2003). The T^2 test was performed using the add-on package ICSNP (version 1.0-7) for R (version 2.10.0, (R Development Core Team 2009)). The standard and high N:COD cases show good agreement between iDynoMiCS and previous models considering the variation between models and the differences in representing the biomass (% deviation < 10%, p-values > 15%). However, results in the low N:COD case are significantly different (p-value < 5%), with mean values differing < 15% between previous models and iDynoMiCS. This may be expected since it is the case that produced the highest variation of model outputs in the original benchmark study, suggesting that

this case is particularly sensitive to idiosyncrasies of models and implementations (Rittmann et al. 2004).

XI Validation of the Individual-based Chemostat Against the ODE Model

In order to test our individual-based chemostat, we simulated microbial growth using a multi-species community composed of COD-oxidising heterotrophs and ammonia-oxidising autotrophs, based on the BM3 benchmark problem (Tables S2, S3, and S4). For simplicity, we ignored the maintenance and inactivation reactions in these tests, and used a lower influent COD concentration (3 mg COD/L rather than 30 mg COD/L). The corresponding deterministic model was simulated and analysed using Matlab (Ordinary Differential Equation solver ode23s), and the deterministic solution obtained this way was compared with the stochastic solution from iDynoMiCS. Solutions from the two simulation methods were compared in the solute and biomass concentrations predicted by each. In all simulations, we used a dilution rate of 0.02 hr^{-1}, and a timestep of 1 hour. For these conditions, the relative error of steady state variables was <2% for solutes, < 1% for heterotrophs, and <15% for autotrophs. The higher discrepancy for the autotrophs is due to there being fewer individuals (usually only 1) compared to the heterotrophs (which usually number in the thousands); when the number of individuals is small, any variation (such as that due to stochastic dilution) is amplified for the population as a whole. The errors are lower when the time-step is reduced (e.g., autotroph error dropped to 9% when using a timestep of 0.5 hour rather than 1 hour), or when the system size is increased to closer approximate the continuous nature of the ODE solution (data not shown); these results indicate that the models' results converge as expected. In addition, comparison of the time series of the stochastic iDynoMiCS runs (2 independent runs) with the deterministic model illustrates rapid convergence of the simulation results as the simulations progress toward the steady state, even after starting with quite different initial biomass values due to the stochastic initialization of the IbM (Figure S6).

XII Implementation Details of the Metabolic Switch Case Study

Included here are further details on the implementation of the case study in the manuscript. The switch algorithm (Figure S7) is implemented with two agent states, each of which has a different set of active metabolic pathways. When conditions are suitable for the agent to switch its metabolism, a switch request is made, but the lag period must pass before the change of active metabolisms actually occurs. When a switch is finally made, the pathways associated with the current switch state are deactivated, and the pathways associated with the new switch state are activated. Any pathways specified outside the switch description are not affected in this process. Note that the behaviour of the switch is identical when the switch threshold is crossed from either direction, just that

the action on the pathways will be opposite. In the event that the switch condition reverses while the agent is within the lag period, the switch request is cancelled and when conditions are right for a switch the process must restart.

In this study, the idealized wastewater treatment environment has a bulk compartment containing COD, dissolved oxygen, and nitrate (for simplicity, we use nitrate directly rather than using influent ammonium and aerobic nitrifiers as the nitrate source). The 0.2 mg O_2/L threshold concentration (Ye et al. 1995) has been used as the crossover concentration in a previous model of O_2-inhibited denitrification to study the feasibility of simultaneous nitrification and denitrification in membrane reactors (Matsumoto et al. 2007). Because of mutually exclusive gene expression, either aerobic or anaerobic pathways are active at any one time, but not both.

In the cost-for-induction cases, we calculate the maintenance rates as follows. If we invert the induction lag values we obtain rates of induction, which are 1, 0.33, and 0.2 h^{-1} for the Lag-1, Lag-3, and Lag-5 species, respectively. We then make the assumption that 50% of cellular maintenance is due to mRNA and protein turnover, and that for species with a shorter induction lag the protein turnover rate should be higher because it is increased turnover that enables faster induction. Using the Lag-5 maintenance rate as a base for comparison, we compute maintenance rates for the other species with an expression that increases the protein-associated component of the maintenance rate in proportion to the ratio of the induction rates:

$$b_{HX}^{m} = \left(f_p \frac{r_X}{r_5} + \left(1 - f_p\right) \right) \cdot b_{H5}^{m} . \qquad \text{(XII-1)}$$

Here b_{HX}^{m} is the maintenance rate for a generic heterotrophic Lag-X species, f_p is the fraction of maintenance that is protein-associated (assumed to be 50%), r_X is the induction rate for a generic Lag-X species, and b_{H5}^{m} and r_5 are the maintenance and induction rates for the Lag-5 species, respectively. This expression yields the maintenance rates used in this study for the cost-for-faster-induction case: 0.0176 and 0.0399 h^{-1} for the Lag-3 and Lag-1 species, respectively; the Lag-3 maintenance rate is 32% higher than for the no-cost case, and the Lag-1 rate is 3 times higher.

During model initialization, we inoculated a model biofilm with 10 agents of each species evenly spaced on the substratum, with each species forming a contiguous group covering 1/3 of the space.

Also included with this supplementary text are videos showing the growth of the biofilm in each of the cost and no-cost switch cases shown in Figure 7, along with a video showing growth of a biofilm in 3D that was illustrated in Figure 8.

XIII Chemostat Results for Metabolic Switch Case

Simulations to test for mutual invasibility were carried out to determine whether coexistence is possible or which strategy is more competitive (3 replicates).

If the cost of switching is included, the faster changing environments (2, 4, 8 hourly pulses) favour the fastest adapting strategy Lag1, while the slower changing environments (16, 32 hourly pulses) favour the slower adapting strategy Lag3, and the constant environments (aerobic or anaerobic) favour the slowest adapting strategy Lag5 (Table S6).

If switching is assumed to be free of costs, the fastest adapting strategy Lag1 dominates in all fluctuating environments, while in the two constant environments, none of the strategies has any advantage, hence the results are entirely due to chance (stochastic dilution leading to complete washout of the invader in some runs but not in others) (Table S6).

XIV iDynoMiCS Class Structure

Entire projects with many different simulations may be performed without any programming of iDynoMiCS because all parameters of the system, the solutes, and the microbes can be read in from a parameter file (the protocol file) in the eXtensible Markup Language (XML) format. Programming is only necessary when new boundary conditions, new kinetic equations, or species with new behaviour are to be used. In that case, it is useful to know that iDynoMiCS is written entirely in the easy-to-program and well-known object-oriented programming language Java, which also runs on virtually any operating system. The object-oriented structure means that each major piece of the model is self-contained and may easily be updated or replaced with a similar class that offers different functionality; for example, one might wish to replace the default multi-grid solute solver with a new solver that offers different functionality. In addition to enabling modularity, the object-oriented structure also allows more complex classes to derive from simpler ones that do not require the complexity. For example, agents in iDynoMiCS are made up of a hierarchy that adds increasing complexity at each stage: first existence, then component masses, then reactions, then location and size, and finally any species-specific behaviour. Because of this structure, adding agents to iDynoMiCS with new types of behaviour rather than just different parameters simply requires creating a new subclass from a similar, already-existing agent type. Other class types that exhibit an inheritance hierarchy include: reactions, detachment functions, spatially-varying concentrations, and the boundary conditions. Previous IbMs have also used a similar object-oriented design, but one major difference between iDynoMiCS and previous models is that iDynoMiCS contains separate classes to describe reactions rather than encapsulating reactions within species classes; this new structure allows for the flexibility required to carry out the metabolic switch example in the manuscript.

XV Obtaining and Using iDynoMiCS

The iDynoMiCS source code will be made available upon publication of this article at the project website after registration of the user (http://www.idynomics.org). The base software package may also be obtained by contacting one of the authors of this manuscript. Future software improvements and developments will be released similarly, though major improvements may be released only after publication in a scientific journal, if appropriate.

iDynoMiCS is free software released under a GPL-like CeCILL license that allows modification and redistribution of the software under the terms of the license (http://www.cecill.info/index.en.html). There is no commercial support provided for users of iDynoMiCS, nor is guaranteed support of any kind provided. There will, however, be a user mailing list to assist with answering questions, and a tutorial and some other materials will be available on the project website. Also, we intend to run a one-week course introducing the software every year. The authors welcome further development by any interested parties, and 3rd-party submissions may be added to the code base upon review.

XVI References

Alpkvist E, Picioreanu C, van Loosdrecht MCM, Heyden A. 2006. Three-Dimensional Biofilm Model With Individual Cells and Continuum EPS Matrix. Biotech. Bioeng. 94(5):961-979.

Anderson TW. 2003. An Introduction to Multivariate Statistical Analysis. New York: Wiley.

Gear CW. 1971. Numerical initial value problems in ordinary differential equations. Upper Saddle River, NJ, USA: Prentice Hall.

Klapper I, Rupp CJ, Cargo R, Purvedorj B, Stoodley P. 2002. Viscoelastic Fluid Description of Bacterial Biofilm Material Properties. Biotech. Bioeng. 80(3):289-296.

Makoto M, Takuji N. 1998. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Trans. Model. Comput. Simul. 8(1):3-30.

Matsumoto S, Terada A, Tsuneda S. 2007. Modeling of membrane-aerated biofilm: Effects of C/N ratio, biofilm thickness and surface loading of oxygen on feasibility of simultaneous nitrification and denitrification. Biochem. Eng. J. 37(1):98-107.

Noguera D, Picioreanu C. 2004. Results from the multi-species Benchmark Problem 3 (BM3) using two-dimensional models. Wat. Sci. Tech. 49(11-12):169-176.

Peters AC, Wimpenny JWT. 1988. A constant-depth laboratory model film fermentor. Biotech. Bioeng. 32(3):263-270.

R Development Core Team. 2009. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.

Rittmann BE, Schwartz AO, Eberl H, Morgenroth E, Pérez J, van Loosdrecht MCM, Wanner O. 2004. Results from the multi-species Benchmark Problem 3 (BM3) using one-dimensional models. Wat. Sci. Tech. 49(11-12):163-168.

Sethian JA. 1999. Fast Marching Methods. SIAM Review 41(2):199-235.

Shampine LF. 1982. Implementation of Rosenbrock methods. ACM Trans. Math. Software 8(2):93-113.

Shampine LF, Reichelt MW. 1997. The Matlab ODE Suite. SIAM J. Sci. Computing 18(1):1-22.

Wanner O, Gujer W. 1986. A multispecies biofilm model. Biotech. Bioeng. 28:314-328.

Ye RW, Haas D, Ka JO, Krishnapillai V, Zimmermann A, Baird C, Tiedje JM. 1995. Anaerobic activation of the entire denitrification pathway in *Pseudomonas aeruginosa* requires Anr, an analog of Fnr. J. Bacteriology 177(12):3606.

XVII Tables

Table S1: Simple example reaction description and stoichiometric matrix

Reaction	Solutes		Biomass			Kinetic Expression
	S_{COD}	S_{O2}	X_A	X_{EPS}	X_I	
Growth	$-\frac{1}{Y_H}$	$-\frac{1-Y_H}{Y_H}$	$1-Y_E$	Y_E		$r_1 = \gamma_1 X_A = \mu^{\max} \frac{S_{COD}}{K^S_{COD}+S_{COD}} \frac{S_{O2}}{K^S_{O2}+S_{O2}} X_A$
Maintenance		-1	-1			$r_2 = \gamma_2 X_A$
Decay			-1		1	$r_3 = \gamma_3 X_A$
EPS release				-1		$r_4 = \gamma_4 X_{EPS}$

Table S2: Reactions matrix for BM3 problem (Rittmann et al. 2004)

Reaction		Solutes			Biomass			Kinetic Expression
		S_{COD}	S_{O2}	S_N	X_H	X_A	X_I	
Heterotroph	*Growth*	$-\frac{1}{Y_H}$	$-\frac{1-Y_H}{Y_H}$		1			$\mu_H^{\max} \frac{S_{COD}}{K^H_{COD}+S_{COD}} \frac{S_{O2}}{K^H_{O2}+S_{O2}} X_H$
	Maintenance		-1		-1			$b_H^m \frac{S_{O2}}{K^H_{O2}+S_{O2}} X_H$
	Decay				-1		1	$b_H^i X_H$
Autotroph	*Growth*		$-\frac{4.57-Y_A}{Y_A}$	$-\frac{1}{Y_A}$*	1			$\mu_A^{\max} \frac{S_N}{K^A_N+S_N} \frac{S_{O2}}{K^A_{O2}+S_{O2}} .X_A$
	Maintenance		-1		-1			$b_A^m \frac{S_{O2}}{K^A_{O2}+S_{O2}} X_A$
	Decay				-1		1	$b_A^i X_A$

*Note: the coefficient on S_N for the autotrophic growth process is corrected from the incorrect coefficient given in (Rittmann et al. 2004).

Table S3: Parameter values for BM3 problem (Rittmann et al. 2004)

Parameter			Value	
Heterotroph	Maximal growth-rate	$\mu_H^{\max}$	0.25	h^{-1}
	Saturation constant for COD	K_{COD}^H	$4 \cdot 10^{-3}$	$gCOD \cdot L^{-1}$
	Saturation constant for O_2	K_{O2}^H	$0.2 \cdot 10^{-3}$	$gO_2 \cdot L^{-1}$
	Maintenance rate	b_H^m	0.0133	h^{-1}
	Inactivation rate	b_H^i	0.0033	h^{-1}
	Biomass yield	Y_H	0.63	gCOD-X / gCOD-S
Autotroph	Maximal growth-rate	$\mu_A^{\max}$	0.0417*	h^{-1}
	Maintenance rate	b_A^m	0.005	h^{-1}
	Inactivation rate	b_A^i	0.00125	h^{-1}
	Saturation constant for N	K_N^A	$1 \cdot 10^{-3}$*	$gN \cdot L^{-1}$
	Saturation constant for O_2	K_{O2}^A	$0.5 \cdot 10^{-3}$	$gO_2 \cdot L^{-1}$
	Biomass yield	Y_A	0.24*	gCOD-X/gN

*Note: these values in (Rittmann et al. 2004) were published erroneously. We have used the values given in the table, which were those actually used in the previous BM3 simulations.

Table S4: Environmental conditions for BM3 problem (Rittmann et al. 2004)

Scenario	Standard	High N:COD	Low N:COD
COD_{in} [g COD/m^3]	30	30	30
NH_{4in}^+ [g N/m^3]	6	30	1.5
O_2 [g COD/m^3]	10	10	10
D [$hour^{-1}$]		0.67	
σ_R [m^2/m^3]		80	
Biomass Density [g COD/L]		15*	
Thickness [μm]		500	

*Note: this value is corrected from the parameter values given in (Rittmann et al. 2004).

Table S5: Results of BM3 simulations

Model Name	Standard Case		High N:COD		Low N:COD	
	COD [mg/L]	NH_4^+ [mg N/L]	COD [mg/L]	NH_4^+ [mg N/L]	COD [mg/L]	NH_4^+ [mg N/L]
CP	5.14	1.50	5.45	18.15	4.39	0.44
DN	5.14	1.74	5.56	20.26	4.98	0.48
W (1D)	5.39	1.59	5.86	18.93	5.19	0.48
M1 (1D)	4.84	1.45	5.35	17.03	4.66	0.45
Average of previous	5.13	1.57	5.56	18.59	4.81	0.46
Std dev. of previous	0.23	0.13	0.22	1.36	0.35	0.02
% variation	4.39	8.11	3.97	7.30	7.33	4.46
iDynoMiCS	5.23	1.46	5.74	17.3	5.05	0.53
% deviation of iDynoMiCS from above average	2.00	7.01	3.33	6.95	5.10	14.59
p-values from T2 test	0.3033		0.1796		0.0082	

The previous models are taken from the IWA BM3 results (Noguera and Picioreanu 2004), and the models included are CP (a particle-based model), DN (a cellular automata model), W (a 1D continuum model), and M1 (a 1D continuum model).

Table S6: Chemostat results for the metabolic switch case study

With cost

Aerobic: Lag5 dominant strategy

Resident \ Invader	Lag1	Lag3	Lag5
Lag1	X	3/3	3/3
Lag3	0/3	X	3/3
Lag5	0/3	0/3	X

Anaerobic: Lag5 dominant strategy

Resident \ Invader	Lag1	Lag3	Lag5
Lag1	X	3/3	3/3
Lag3	0/3	X	3/3
Lag5	0/3	0/3	X

Pulse 2h: Lag1 dominant strategy

Resident \ Invader	Lag1	Lag3	Lag5
Lag1	X	0/3	0/3
Lag3	2/3	X	0/3
Lag5	3/3	0/3	X

Pulse 4h: Lag1 dominant strategy

Resident \ Invader	Lag1	Lag3	Lag5
Lag1	X	0/3	0/3
Lag3	3/3	X	0/3
Lag5	3/3	3/3	X

Pulse 8h: Lag1 best strategy

Resident \ Invader	Lag1	Lag3	Lag5
Lag1	X	0/3	0/3
Lag3	0/3	X	0/3
Lag5	3/3	3/3	X

Pulse 16h: Lag3 dominant strategy

Resident \ Invader	Lag1	Lag3	Lag5
Lag1	X	0/3	0/3
Lag3	0/3	X	0/3
Lag5	3/3	3/3	X

Pulse 32h: Lag3 dominant strategy

Resident \ Invader	Lag1	Lag3	Lag5
Lag1	X	0/3	0/3
Lag3	0/3	X	0/3
Lag5	3/3	3/3	X

Without costs

Aerobic: Neutral fitness

Resident \ Invader	Lag1	Lag3	Lag5
Lag1	X	1/3	2/3
Lag3	2/3	X	0/3
Lag5	3/3	0/3	X

Anaerobic: Neutral fitness

Resident \ Invader	Lag1	Lag3	Lag5
Lag1	X	1/3	0/3
Lag3	1/3	X	1/3
Lag5	2/3	2/3	X

Pulse 2h: Lag1 dominant strategy

Resident \ Invader	Lag1	Lag3	Lag5
Lag1	X	1/3	0/3
Lag3	1/3	X	1/3
Lag5	2/3	2/3	X

Pulse 4h: Lag1 dominant strategy

Resident \ Invader	Lag1	Lag3	Lag5
Lag1	X	0/3	0/3
Lag3	3/3	X	0/3
Lag5	3/3	2/3	X

Pulse 8h: Lag1 dominant strategy

Resident \ Invader	Lag1	Lag3	Lag5
Lag1	X	0/3	0/3
Lag3	3/3	X	0/3
Lag5	3/3	2/3	X

Pulse 16h: Lag1 dominant strategy

Resident \ Invader	Lag1	Lag3	Lag5
Lag1	X	0/3	0/3
Lag3	3/3	X	0/3
Lag5	3/3	3/3	X

Pulse 16h: Lag1 dominant strategy

Resident \ Invader	Lag1	Lag3	Lag5
Lag1	X	0/3	0/3
Lag3	3/3	X	0/3
Lag5	3/3	3/3	X

- While number of agents moving is greater than 5%:
  - For each agent P_i:
    - For each agent P_j in the neighbourhood of P_i:
      - Compute the agent-agent overlap distance δ_{ij}.
      - Store a movement of $\delta_{ij}/2$ for agent P_i in the direction away from P_j (add to stored movement).
      - Store a movement of $\delta_{ij}/2$ for agent P_j in the direction away from P_i (add to stored movement).
    - End
  - End
  - For each agent P_i:
    - If the stored movement is nonzero:
      - Apply the stored movement.
      - Increment the count of agents moved.
    - End
  - End
- End

Algorithm S1: Pseudo-code describing the procedure for agent shoving.

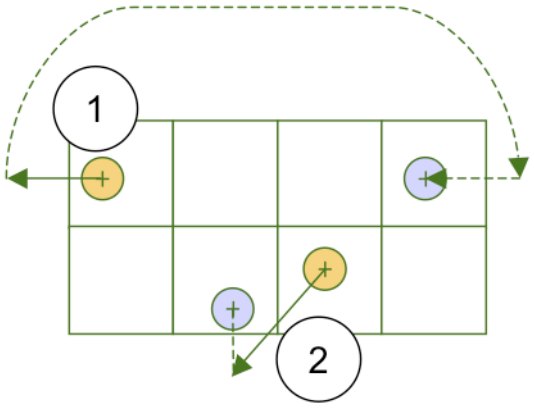

Figure S1: Agent behaviour at boundaries

Shown is the behaviour of located agents crossing a (1) periodic and a (2) zero-flux boundary. Solid arrows represent the initial movement, dashed arrows the corrected movement.

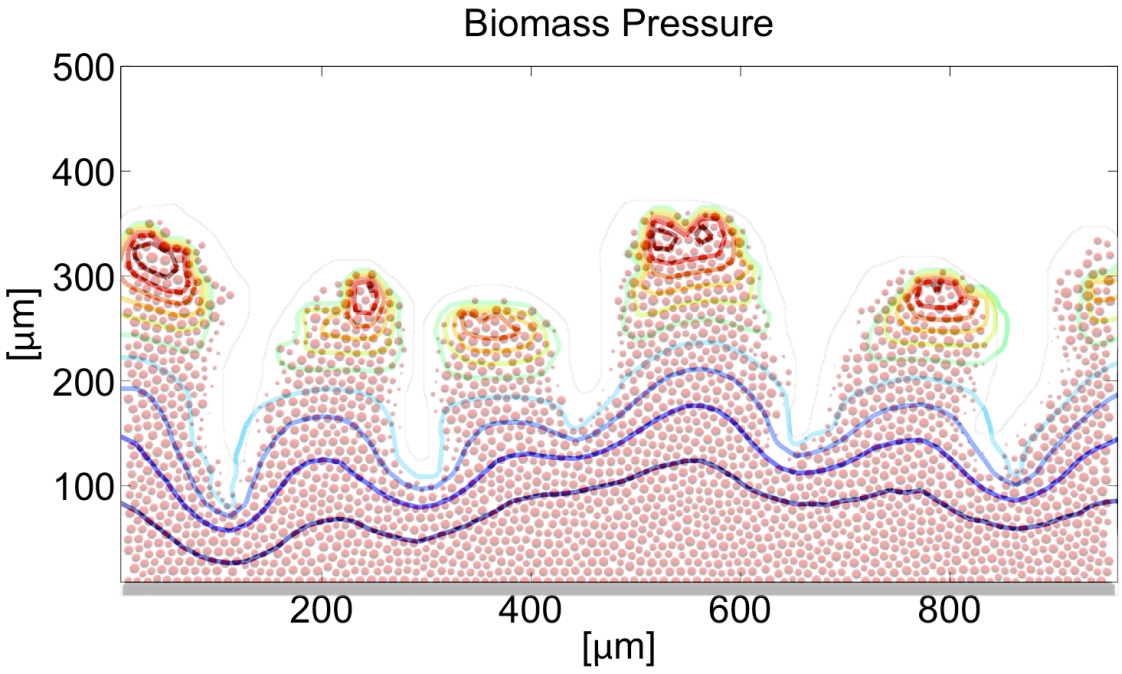

Figure S2: Pressure field contours

The red and yellow contours near the finger tips represent positive pressure leading to expansion, while the darker blue contours in the depths of the biofilm represent negative pressures leading to contraction.

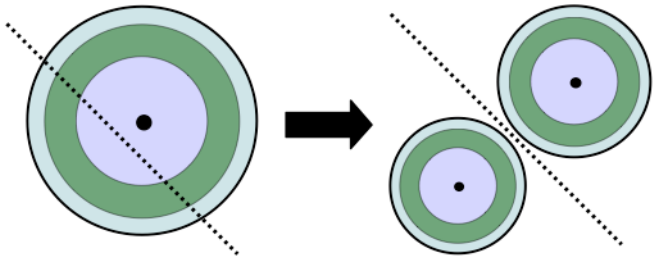

Figure S3: Cell division.

Cell division with stochastic deviation from symmetric cell division yields unequal daughter cell volumes. There is also stochastic selection of the division direction.

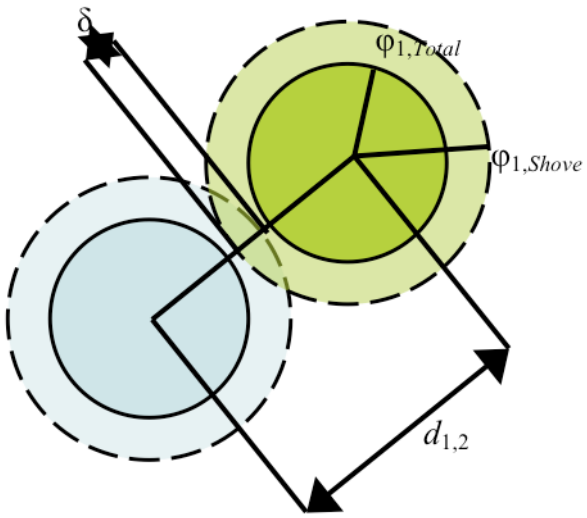

Figure S4: Shoving of neighbouring cells 1 and 2 is triggered if there is an overlap between them.

Overlap distance between agents is δ, distance between cell centres is $d_{1,2}$, total agent radius is $\varphi_{j,Total}$, and shoving radius is $\varphi_{j,Shove} = k_{Shove} \cdot \varphi_{j,Total}$.

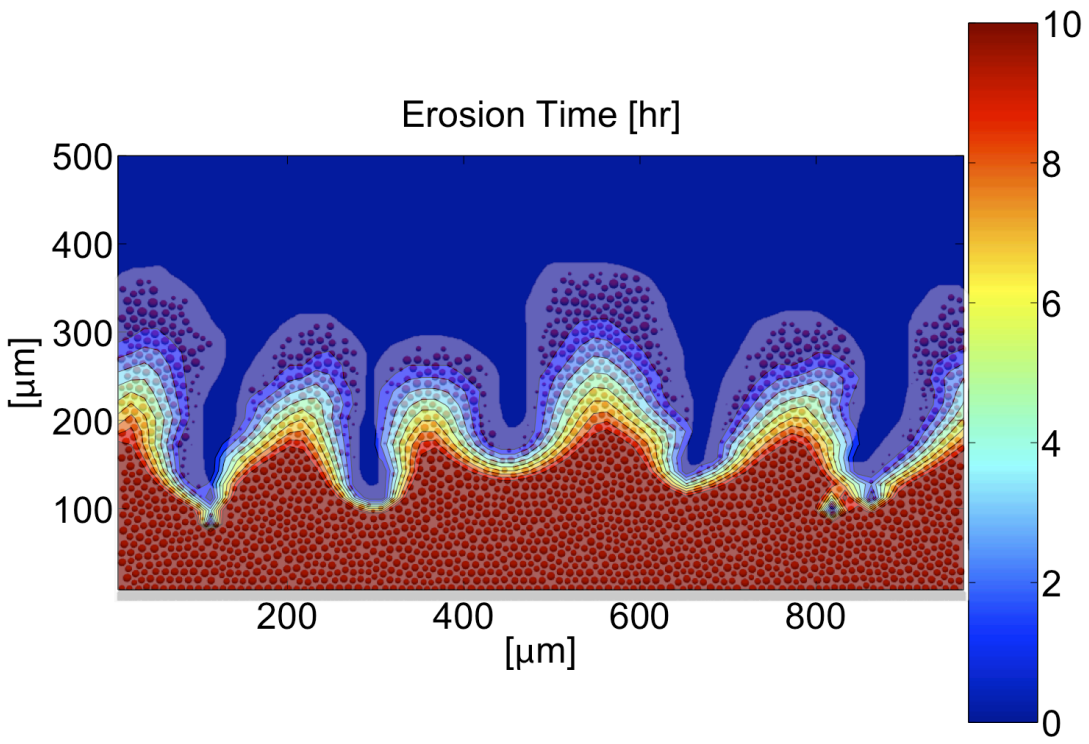

Figure S5: Example of isolines of erosion time computed with the fast marching algorithm

The plotted erosion strength is a quadratic function of height with a high strength of $k_{Det} = 200\left(\mu m \cdot h\right)^{-1}$, and the isolines represent the batch of agents that will be removed after an integer number of hours. This high erosion strength will remove a high amount of biomass very quickly.

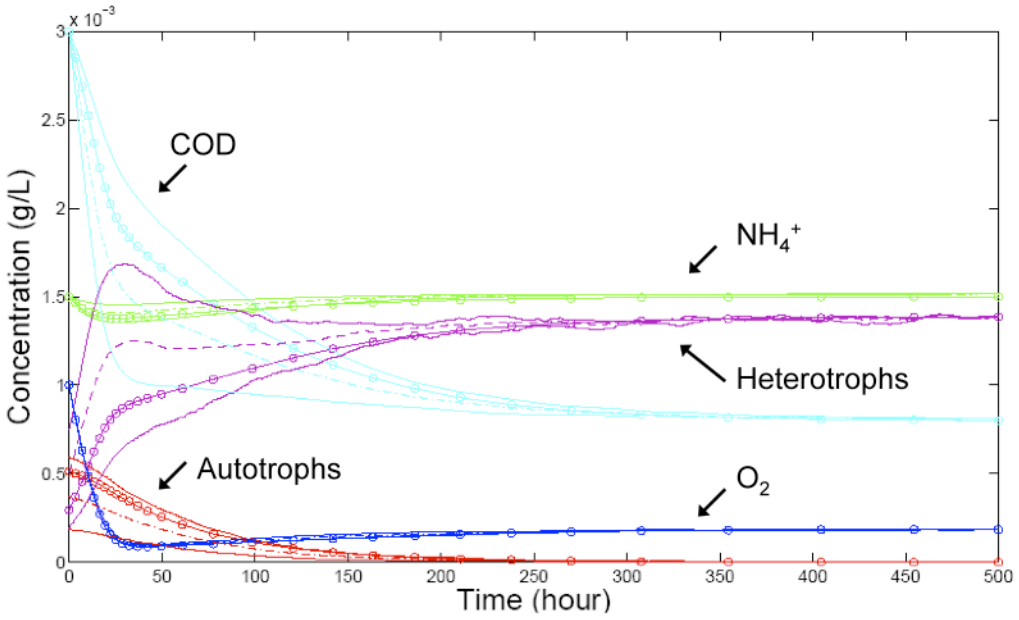

Figure S6: Comparison of simulations of the stochastic chemostat model using iDynoMiCS with simulations of the deterministic ODE model using the ODE solver ode23s of Matlab.

Circles indicate the deterministic solution, the two solid lines correspond to two of the three iDynoMiCS simulations performed; these two envelope the third replicate The dashed line represents the means of each time point of the 3 replicates. Growth parameters were as in BM3 (except maintenance and inactivation reactions, which were not considered in this analysis) and environmental conditions were as in the Low N:COD Scenario, but using a COD concentration of 3 mg COD/L.

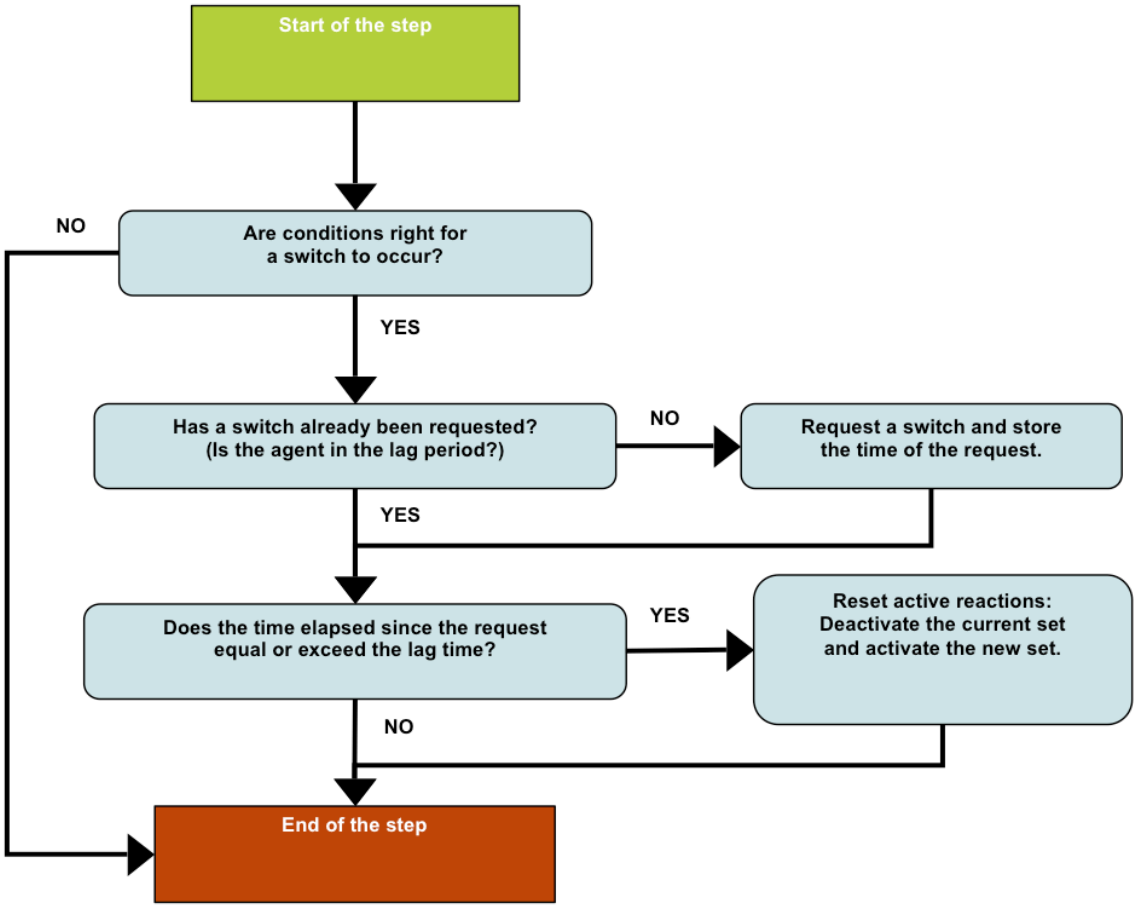

Figure S7: Metabolic switch algorithm

The behaviour is identical for any type of switching (activating or deactivating pathways). First the conditions for a new pathway must be met (in our example, the oxygen concentration must have crossed the threshold), then a time equal to the lag period must elapse before the switch actually occurs. If the conditions change such that a switch is not appropriate, the request is cancelled and the process restarts.

XIX Movies

Movies S1-S8: These movies illustrate biofilm growth *without cost* for fast switching.

Movie S1: Growth under aerobic conditions.
Movie S2: Growth under anaerobic conditions.
Movie S3: Growth under anaerobic conditions interrupted by an oxygen pulse every 32 hours.
Movie S4: Growth under anaerobic conditions interrupted by an oxygen pulse every 16 hours.
Movie S5: Growth under anaerobic conditions interrupted by an oxygen pulse every 8 hours.
Movie S6: Growth under anaerobic conditions interrupted by an oxygen pulse every 4 hours.
Movie S7: Growth under anaerobic conditions interrupted by an oxygen pulse every 2 hours.
Movie S8: Growth under anaerobic conditions interrupted by randomly-occurring oxygen pulses.

Movies S9-S16: These movies illustrate biofilm growth *when there is a cost* for fast switching.

Movie S9: Growth under aerobic conditions.
Movie S10: Growth under anaerobic conditions.
Movie S11: Growth under anaerobic conditions interrupted by an oxygen pulse every 32 hours.
Movie S12: Growth under anaerobic conditions interrupted by an oxygen pulse every 16 hours.
Movie S13: Growth under anaerobic conditions interrupted by an oxygen pulse every 8 hours.
Movie S14: Growth under anaerobic conditions interrupted by an oxygen pulse every 4 hours.
Movie S15: Growth under anaerobic conditions interrupted by an oxygen pulse every 2 hours.
Movie S16: Growth under anaerobic conditions interrupted by randomly-occurring oxygen pulses.

Movie S17: This movie illustrates a 3D simulation of biofilm growth under anaerobic conditions interrupted by an oxygen pulse every 4 hours when there is a cost for fast switching.