BioScape: A Modeling and Simulation Language for Bacteria-Materials Interactions

Adriana Compagnoni[a] Vishakha Sharma[a] Yifei Bao[a]
Matthew Libera[a] Svetlana Sukhishvili[a] Philippe Bidinger[b]
Livio Bioglio[c] Eduardo Bonelli[d]

[a] *Stevens Institute of Technology*
[b] *VERIMAG*
[c] *Università di Torino*
[d] *Universidad Nacional de Quilmes*

Abstract

We design BioScape, a concurrent language for the stochastic simulation of biological and bio-materials processes in a reactive environment in 3D space. BioScape is based on the Stochastic Pi-Calculus, and it is motivated by the need for individual-based, continuous motion, and continuous space simulation in modeling complex bacteria-materials interactions. Our driving example is a bio-triggered drug delivery system for infection-resistant medical implants. Our models in BioScape will help in identifying biological targets and materials strategies to treat biomaterials associated bacterial infections.

The novel aspects of BioScape include syntactic primitives to declare the scope in space where species can move, diffusion rate, shape, and reaction distance, and an operational semantics that deals with the specifics of 3D locations, verifying reaction distance, and featuring random movement. We define a translation from BioScape to 3π and prove its soundness with respect to the operational semantics.

Keywords: Stochastic simulation, stochastic Pi-calculus, bacteria-materials interaction

1 Introduction

In contrast to the now deep and multidimensional understanding of how tissue cells interact with the surface of biomaterials, comparatively little is known about the influence of surface properties on interactions with bacteria. These interactions are clearly very important, however. Biomaterials-associated infection (BAI) is a major clinical problem [1,13,35]. Current strategies to mitigate BAI concentrate on engineering antimicrobial [16,18,20,22] or antifouling [9,14,19,33] coatings for specific biomedical devices.

While valuable clinically in the short-term, this approach ignores the fact that many devices, particularly those involving regenerative strategies, require surfaces that must controllably interact with both tissue cells and bacteria. Often, surfaces

optimized to promote tissue-cell interactions also support undesirable bacterial colonization (Fig. 1).

Concurrent Modeling of Biomaterial Interactions

As an alternative to models built around sets of ordinary differential equations (ODEs), process algebras are formal languages where multiple objects with different behavioral attributes can interact with each other and dynamically influence overall system development. Process algebras are being used to model biological systems [30,6,2], where they are particularly attractive, because of their ability to accommodate new objects and new behavioral attributes as the complex biological system becomes better understood.
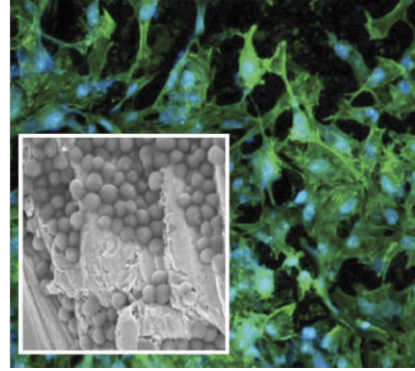

Fig. 1 Grit-blasted Ti6Al4V used in orthopedic implants promotes osteoblast adhesion/spreading (fluorescence image) but also enhances staphylococcal colonization (SEM inset). After Wu, Libera et al. [37]

Currently, however, modeling languages based on concurrent synchronization either lack spatial attributes (SPiM [27], Kappa [12], Petri Nets [26]) or stochasticity (SpacePi [15], Shape Calculus [3]), or they offer only a limited notion of space (BioAmbients [23], BioPepa [10]). As findings from biological experiments reveal, inter and intracellular dynamics and signaling pathways depend on the location and movement of particles [17]. Recently, new spatial modeling languages allowing explicit description of temporal spatial dynamics of biochemical processes have been proposed (SpacePi [15], DCA [36], $L\Pi$ [34], Stochsim [24]). Other agent-based platforms [21] include C-Immsim
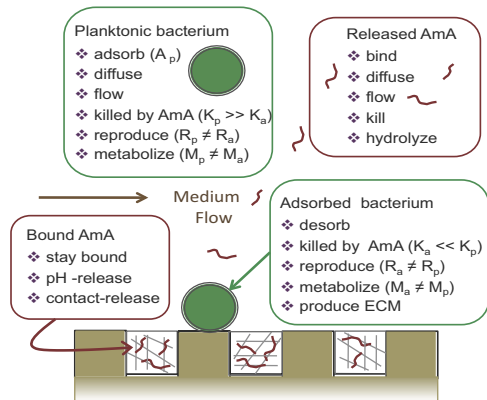

Fig. 2 Concurrency models flexibly account for multiple object types and copies. Each type has characteristic attributes. Expression of one attribute dynamically influences other objects and what attribute each might next express.

[32,8] and PathSim visualizer [28]. However, few of them support individual based, continuous motion, and continuous space stochastic simulation [4], which are important features for modeling temporal spatial dynamics of biochemical processes accurately. To address this problem we design BioScape, a new language incorporating both stochasticity and 3D spatial attributes.

Fig. 2 illustrates the elements of a simple instance of the model. It includes four objects: planktonic bacteria; adsorbed bacteria; gel-bound antibacterial agent

(AmA); and released AmA. Each has attributes, many of which are inter-related. Both bacteria, for example, can metabolize, albeit at different rates. Metabolism is particularly significant for bound AmA, since acidic products will lower the local pH, reduce the strength of electrostatic AmA-gel binding, and enable the pH-release of bound AmAs. Released AmA can then diffuse and/or flow, and potentially kill bound or planktonic bacteria, again with different efficiencies, because of the enhanced microbial resistance of biofilm bacteria. The complexity of such a model can rapidly explode as more attributes are assigned to each object. Note that spatial coordinates and stochasticity are critical to understanding the interactions between different objects and with the substrate, because they all rely on proximity. So far, existing process-algebra modeling languages do not support this. Furthermore, while there are now a number of models being developed to explain the biofilm formation [31,38], none takes into account the controllable properties of the substrate and how this can influence bacterial adhesion, proliferation, and phenotypic change.
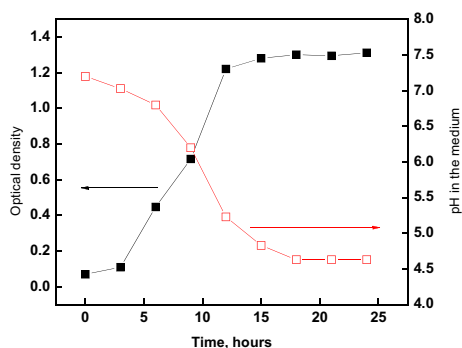

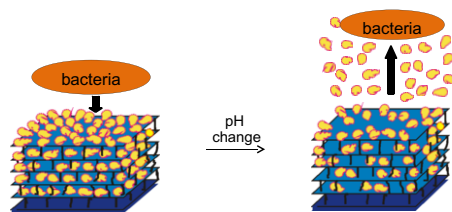Fig. 3 pH variation during S. epidermidis growth

Fig. 4 Bio-triggered release of antibacterial agents from pH-responsive layer-by-layer hydrogel thin film.

A number of experimentally measurable quantities can be used to calibrate elements of the model. Among these are the bacterial proliferation rates, the biofilm morphology, the AmA minimum inhibitory (MIC) and bactericidal (MBC) concentrations, gel capacities for AmA binding and pH-dependent properties, and component diffusivities, among others. For example, measurements by Sukhishvili's group of the pH-dependent release of L5 antimicrobial peptide and its effect on S. epidermidis [25] have been used by Compagnoni and her group to develop a prototype implementation in BioScape (Fig. 3). This data corresponds to the schema described in Fig. 4. This model includes computational processes for planktonic bacterial motion, adhesion, and proliferation on an L5-loaded hydrogel surface as well as for the local metabolic pH decrease, triggered AmA release, and killing of bacteria (Figs. 5 C/D). Experimental images (Figs. 5 A/B) are in good agreement with our computational results.

Language Design

The 3D aspect of BioScape is inspired by 3π [7] and SpacePi [15]. In BioScape every process has an implicit affine map and a shape (Fig. 7). The shape is defined by a set of point coordinates that is local to a process' system of coordinates. While the syntax of BioScape describes processes within its implicit local frame, its semantics places processes in a global 3D space (Fig. 8), by assigning an affine map to each process – a located process. The application of an affine map to the shape returns the shape located in the global frame. The

Fig. 5 Optical micrographs of NJ 9709 S. epidermidis cultured on (A) as-synthesized and (B) L5-loaded (PMAA)10 gels. BioScape simulations with 40 bacteria and (C) an as-synthesized gel film and (D) an L5-loaded gel film. Live bacteria are green, dead bacteria are red, and L5 molecules are small blue dots.

affine map is what characterizes the position of the process. Movement is then modeled by updating a process' affine map. On the contrary, 3π is a low level language that gives absolute control of spatial attributes to the programmer. For example, the programmer can guard an interaction by checking whether two processes are close enough, and it can also assign affine maps to processes. For example, collision checks in 3π would have to be implemented by the programmer checking corresponding distances with every other entity in the system, and taking into account the shape of every process. In contrast, in BioScape, the programmer specifies species declaring a reaction radius, as in SpacePi, and the operational semantics enforces the proximity requirement. Therefore, affine maps are not available to the programmer in BioScape, unlike 3π. While 3π is a general calculus for processes in space, BioScape has been designed to program biological and biomaterial processes and their interactions.

New Features of BioScape

- We introduce three new parameters for each process: movement space, movement step, and shape to be specified by the programmer when defining a process. We take the definition of bacteria as an example. The movement space restricts the space within which bacteria can move. In Fig. 5, bacteria can never penetrate the gel film. Thus we define the volume above the film as the movement space for bacteria. We assume that the scheduler will randomly place the initial concentration of bacteria in that space. The movement step indicates the distance that bacteria can move in a time interval. The shape of bacteria represents the volume that each bacterium occupies in space. The shape is instrumental in avoiding collisions and overlaps.
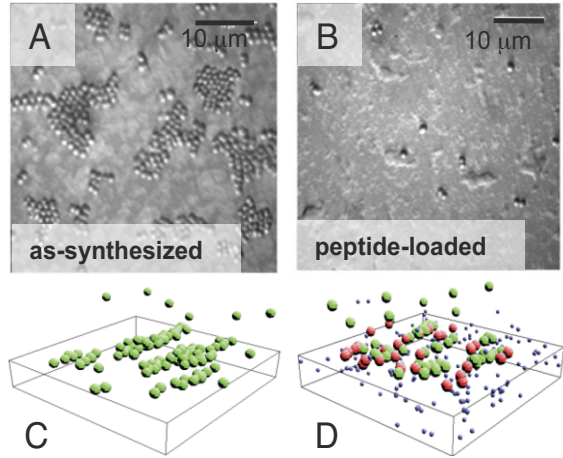
- Besides standard reaction operations send, receive and delay, BioScape has a geometric operation move, that can be generalized to any affine transformation. The difference is that while reaction operations have stochastic reaction rates, the operation move is always enabled. In order to avoid an unfair competition between reactions and geometric operations, we split the semantics into two reduction relations, and we consider a heterogeneous choice operator with probabilistic and non-deterministic branches. Intuitively, it is like having two choice operators.

- The design decisions of having the scheduler assign initial affine maps to each process and of having the operational semantics enforcing proximity requirements significantly simplifies the models' code.

BioScape is a collaboration tool that has helped us develop a long term multidisciplinary research program to study how surface properties affect interactions with bacteria.

2 Syntax

BioScape is based on the stochastic π-calculus [29] with primitives for processes in 3D space. We assume a set of names N ranged over by $x, y, \ldots$, and a global three-dimensional Euclidean space. The syntax of BioScape is defined in Fig. 6. $(\nu x@\mathtt{r},\mathtt{rad}).P$ defines channel name x with two parameters $\mathtt{r}$ and $\mathtt{rad} \in \mathbb{R}_{\geq 0}$ in process P; $\mathtt{r}$ is the stochastic rate for communications through channel x and $\mathtt{rad}$ is the communication radius. The radius is the maximum distance between processes in order to communicate through chan-

$$P, Q ::= (\nu x@\mathtt{r},\mathtt{rad}).P \quad | \quad P \,|\, Q \quad | \quad X(x)$$
$$M ::= \mathbf{0} \quad | \quad \pi.P + M$$
$$\pi ::= \mathtt{delay@r} \quad | \quad !x(x) \quad | \quad ?x(y) \quad | \quad \mathtt{mov}$$
$$D ::= \emptyset \quad | \quad D, X(x)@\xi, \omega, \sigma = M$$
$$E ::= \emptyset \quad | \quad E, x@\mathtt{r},\mathtt{rad}$$

Fig. 6 BioScape Syntax

nel x, and the reaction rate determines whether two processes that are close enough to react actually do. $P \,|\, Q$ is parallel composition of processes. M is the *heterogeneous* choice, where $+$ is associative and commutative. M may have reaction branches and movement branches. The reaction branches are probabilistic, while the movement branches are non-deterministic, since reactions are subject to reaction rates, while movement is always enabled. $\mathbf{0}$ is the empty process. The prefix π denotes the action that the process $\pi.P$ can perform. $\mathtt{delay@r}$ is a spontaneous and unilateral reaction of a single process, and $\mathtt{r}$ is the stochastic rate. $!x$ is the output prefix and $?x$ is the input prefix. We add a new geometry prefix $\mathtt{mov}$ to translate a process. We use standard syntactic abbreviations such as $\pi.P$ for $\pi.P + 0$ and π for $\pi.0$. $X(x)$ is a definition call.

D is a global list of definitions. $X(x)@\xi, \omega, \sigma = M$ defines process X with argument x, movement space ξ, step ω and shape σ. The movement space ξ is a set of point coordinates in the global coordinate system defining a volume. Intuitively, X can move within ξ. The step $\omega \in \mathbb{R}_{\geq 0}$, is the distance X can move in a single step, and it corresponds to the diffusion rate of X; the shape of X is σ, an arbitrary volume in space defined as a set of coordinates in the local coordinate system (Fig. 7). The movement space for the empty process $\mathbf{0}$ is everywhere, the global space, and
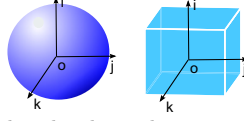
Fig. 7. Shapes (σ) in their local coordinate system (i,j,k) with origin o.
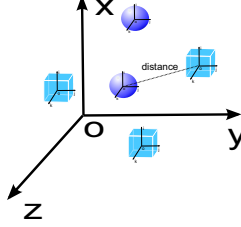
Fig. 8. Shapes in the global coordinate system (x,y,z) – ($\mu(\sigma)$).

NR.MOVE
$$\frac{\mu' = \mathtt{translate}(\omega, \mu, r) \quad \mu'(\sigma) \subseteq \xi \quad X(x)@\xi, \omega, \sigma = \mathtt{mov}.P + M \in D}{E \vdash \{X(y)\}_\mu \to \{P[y/x]\}_{\mu'}}$$

NR.PAR
$$\frac{E \vdash A \to B \qquad \mathtt{trans}(B) \cap \mathtt{trans}(C) = \emptyset}{E \vdash A \mid C \to B \mid C}$$

SR.DELAY
$$\frac{X(x)@\xi, \omega, \sigma = delay@\mathtt{r}.P + M \in D}{E \vdash \{X(y)\}_\mu \to_{\mathtt{r}} \{P[y/x]\}_\mu}$$

SR.COM
$$\frac{X(x)@\xi, \omega, \sigma =!z(u).P + M \in D \quad Y(y)@\xi, \omega, \sigma =?z(v).Q + N \in D \quad \mathtt{dis}(\mu, \mu') \leq \mathtt{rad}}{E, z@\mathtt{r}, \mathtt{rad} \vdash \{X(x')\}_\mu \mid \{Y(y')\}_{\mu'} \to_{\mathtt{r}} \quad \{P[x'/x]\}_\mu \mid \{Q[y'/y][u/v]\}_{\mu'}}$$

Fig. 9. Stochastic (SR) and Non-Stochastic (NR) Reduction Relations (Sample rules)

its movement step is 0 by default. X can be defined by at most one equation in D.

E is an environment of channel name declarations. $x@\mathtt{r}, \mathtt{rad}$ declares channel name x with reaction rate $\mathtt{r}$ and reaction radius $\mathtt{rad}$. A channel name x appears at most once in E.

Consider the following simple example of a bacterium `Bac`, that can either move or divide into two daughter cells. A more complex example can be found in Section 5. `Bac` is defined with movement space `movB`, movement step `stepB`, and shape `shapeB`. Intuitively, bacteria can move within `movB`, with random steps of length `stepB`, and the shape `shapeB` is at all times contained within `movB`. The prefix `mov` represents a random movement of length `stepB`. `delay@1.0.(Bac() | Bac())` represents mitosis, the division of a bacterium into two daughter cells: `Bac() | Bac()`, and the `delay@1.0` prefix is used to model the fact that division is not an instantaneous reaction.

```
Bac()@movB, stepB, shapeB =  mov.Bac() + delay@1.0.(Bac() | Bac())
```

3   Operational Semantics

The operational semantics of BioScape is based on two reduction relations: a non-deterministic relation, $E \vdash_D A \to B$, for geometric transformations, in our case move, and a stochastic relation, $E \vdash_D A \to_{\mathtt{r}} B$, for reactions such as synchronization and delay. We often omit D to simplify the notation.

We use μ to represent an affine map; $\mu(s) = M \times s + V$, where M is a matrix

and V is a vector [7] (see Fig. 7). $\mu(\sigma)$ computes the location and orientation of a process in the global coordinate system. When a process is initialized, it is assigned a random μ within its movement space ξ in the global frame ($\mu(\sigma) \subseteq \xi$).

We define *configurations* $A, B, \ldots$ as parallel compositions of *located processes*. A located process $\{P\}_{\mu,\sigma}$ is a process P annotated with affine map μ and shape σ. $A, B, \ldots ::= \{P\}_{\mu,\sigma} \quad | \quad A \mid B \quad | \quad (\nu x@\mathbf{r},\mathbf{rad}).A$. For clarity, we often omit the shape σ.

Evaluation contexts are given as follows. We write $\mathbf{C}[A]$ for the context $\mathbf{C}$ in which the hole $\cdot$ has been substituted with A: $\mathbf{C} ::= \cdot \quad | \quad (\nu x@\mathbf{r},\mathbf{rad}).\mathbf{C} \quad | \quad A \mid \mathbf{C} \quad | \quad \mathbf{C} \mid A$

We define $\mathtt{trans}$, a function on configurations that returns the shapes of its processes located in the global frame (Fig. 8), such that $\mathtt{trans}(\{P\}_{\mu,\sigma}) = \mu(\sigma)$, $\mathtt{trans}((\nu x@\mathbf{r},\mathbf{rad}).A) = \mathtt{trans}(A)$ and $\mathtt{trans}(A \mid B) = \mathtt{trans}(A) \cup \mathtt{trans}(B)$. We write $\mathtt{dis}(\mu,\mu')$ for the distance between the origin of μ and the origin of μ' in the global frame (Fig. 8). We denote $\mathtt{translate}(\omega,\mu,r)$ the function that generates a new local affine map μ', using the movement step ω, the old map μ, and a random number r.

As usual, $\mathtt{fn}$ is a function that returns the set of free names of a process or a configuration, $\mathtt{bn}$ is a function

(S.Loc)	$P \equiv Q$ implies $\{P\}_\mu \equiv \{Q\}_\mu$
(S.Loc.Nu)	$(\nu x@\mathbf{r},\mathbf{rad}).\{P\}_\mu \equiv \{(\nu x@\mathbf{r},\mathbf{rad}).P\}_\mu$
(S.Loc.Par)	$\dfrac{\mu_1(shape(P)) \uplus \mu_2(shape(Q)) = \mu(shape(P \mid Q))}{\{P \mid Q\}_\mu \equiv \{P\}_{\mu_1} \mid \{Q\}_{\mu_2}}$

Fig. 10 Structural Equivalence (Sample Rules)

that returns the set of bound names of a process or a configuration, and $=_\alpha$ equates two processes or configurations that differ only in their bound names.

We define the shape of processes inductively as follows:

$shape(\mathbf{0}) = \emptyset$ $shape(X(a)) = \sigma$ if $X(x)@\xi, \omega, \sigma = M \in D$
$shape((\nu x@\mathbf{r},\mathbf{rad}).P) = shape(P)$ $shape(P \mid Q) = shape(P) \uplus shape(Q)$

where $\uplus$ gives a shape obtained by composing two shapes through juxtaposition. For different applications we can choose suitable functions to realise $\uplus$, we only require $\uplus$ to be a commutative and associative operator, i.e. $\sigma_1 \uplus \sigma_2 = \sigma_2 \uplus \sigma_1$ and $(\sigma_1 \uplus \sigma_2) \uplus \sigma_3 = \sigma_1 \uplus (\sigma_2 \uplus \sigma_3)$. $\mu(shape(P))$ computes the space occupied by a process P in the global coordinate system.

The structural equivalence, $\equiv$, is the smallest equivalence relation that contains the rules in Fig. 10, and such that parallel composition is commutative, associative, and has neutral element $\{\mathbf{0}\}_\mu$ for any μ. Rule S.Loc uses the standard structural equivalence on processes, Rule S.Loc.Nu allows a channel definition to move through located processes, and Rule S.Loc.Par permits to move from configurations to a single located process (and viceversa) through the creation of a new affine map derived from the old ones and the shape of the processes.

Sample rules for both the non-stochastic ($E \vdash A \rightarrow B$) and stochastic ($E \vdash A \rightarrow_\mathbf{r} B$) reduction relation of BioScape are given in Fig. 9. The condition $\mu'(\sigma) \subseteq \xi$ of NR.Move ensures the new located process $\{P[y/x]\}_{\mu'}$ is within its movement space ξ. NR.Move can be easily generalized to any affine map application. The NR.Par condition $\mathtt{trans}(B) \cap \mathtt{trans}(C) = \emptyset$ means that reduction does not cause collisions

or overlaps. Regarding the stochastic reduction relation, $\mathtt{r}$ in $E \vdash A \rightarrow_{\mathtt{r}} B$ is the rate for synchronization or delay. The condition $\mathtt{dis}(\mu, \mu') \leq \mathtt{rad}$ in SR.Com ensures that located processes $\{X(x')\}_\mu$ and $\{Y(y')\}_{\mu'}$ are close enough to communicate through channel x. The remaining rules are standard.

4 Simulation

The simulation algorithm has two phases: reaction and movement. The reaction phase is based on Gillespie's algorithm as implemented in SPiM, and instead of keeping only concentrations for each agent species, it also keeps 3D information – (x,y,z) – for each instance. Gillespie's algorithm produces two outputs in each iteration: a reaction to be executed next, and a time interval to update the simulation time. If the selected reaction is an interaction between two agents (send/receive) then the algorithm uses 3D location information to identify two individual agents close enough to interact, and proceeds as described in the operational semantics (SR-Com). If there are no two such agents it proceeds to the movement phase. If the selected reaction is a first order reaction (delay), the algorithm propagates 3D information as described in the operational semantics (SR-Delay). The movement phase uses the time interval generated by Gillespie's algorithm and moves each agent in a random direction a distance proportional to the diffusion rate (ω) in that period of time. The movement phase takes into account collision detection as described in the operational semantics (NR-Par).

Overcrowding is beyond the scope of this paper. However there could be several approaches to address it. Cells have external regulators that allow them to react to molecules on the outside of neighboring cells, and those external regulators emit a signal to inhibit mitosis in overcrowded conditions. Therefore, if it were relevant to the system being modeled, the most natural way to address overcrowding would be to program it into the model. Alternatively, the modeling language could address overcrowding in different ways. For instance, if enough movement steps fail within a given movement area in one iteration, reactions exacerbating overcrowding could be disabled. This choice would have the effect of modeling dormancy in cells such as bacteria in Maturation I or Maturation II stage of biofilm formation.

5 Example

In this section, we present the BioScape model for the bio-triggered drug delivery system from Figs. 4 and 5. We first define the communication channels `release,` `kill` and `bind` with reaction rate and reaction radius.

Channel `release` is for the communication between hydronium ions and embedded drug molecules, channel `kill` is for the communication between released drug molecules and bacteria, and channel `bind` is for the communication between bacteria in solution and binding sites on the hydrogel film substrate. `BacF()` represents a free bacterium in solution. Free bacteria can move, bind to the hydrogel film, grow, acidify the environment by producing hydronium ions, and get killed by

drug molecules. `BacB()` represents a bacterium bound to the hydrogel film. Bound bacteria cannot move, but they can grow, acidify the environment by producing hydronium ions, and get killed by drug molecules. `BindSite()` represents a binding site on the hydrogel film. `DeadBac()` represents a dead bacterium. Dead bacteria can move and degrade. `MolB()` represents a drug molecule embedded in the hydrogel film. Drug molecules can be released as hydrogel ions are produced. `MolF()` represents released drug molecules. They can move and kill bacteria. `HIon()` represents a hydronium ion, which determines the pH value of the environment. Hydrogen ions can move and release embedded drug molecules. `resX`, `stepX` and `shapeX` represent ξ, ω and σ for species `X`. The following is the BioScape code.

```
release@0.004, 2.0
kill@0.001, 0.5                            HIon()@resIon, stepIon, shapeIon
bind@3.0, 0.2                                  = mov.HIon() + !release.HIon()

BacF()@resBF, stepBF, shapeBF =            BindSite()@resBS, stepBS, shapeBS
    mov.BacF() +                               = ?bind.BindSite()
    !bind.BacB() +
    delay@0.2.(BacF() | BacF()) +          DeadBac()@stepDB, stepDB, shapeDB
    delay@0.005.(BacF() | HIon()) +            = delay@0.1
    ?kill.DeadBac()
                                           MolB()@resMB, stepMB, shapeMB
BacB()@resBB, stepBB, shapeBB =                = ?release.MolF()
    delay@0.1.(BacB() | BacF()) +
    delay@0.005.(BacB() | HIon()) +        MolF()@resMF, stepMF, shapeMF
    ?kill.DeadBac()                            = mov.MolF() + !kill
```

To simulate this model, the programmer decides on an initial concentration of free bacteria (`BacF()`) and bound molecules of AmA (`MolB()`). The simulation results of Fig. 5 C, show the controlled experiment of assuming only an initial concentration of free bacteria and no antibacterial agent. The results in Fig. 5 D, in contrast, show the effect of AmA molecules on the size and number of bacteria clusters. Figs. 5 A and B show the experimental images consistent with our results.

We now consider an example illustrating the interleaving of stochastic and non-stochastic rules starting from one free bacterium, one hydronium ion and one bound molecule of AmA: $\{\texttt{BacF()}\}_{\mu_1} \mid \{\texttt{HIon()}\}_{\mu_2} \mid \{\texttt{MolB()}\}_{\mu_3}$, to showcase the interleaving of movement steps with reaction steps (communication or delay). To make our example easier to follow, we unfold all three definitions as follows:

$$\{\texttt{mov.BacF()} + \texttt{!bind.BacB()} + \texttt{delay@1.5.(BacF() | BacF())}$$
$$+\texttt{delay@0.5.(BacF() | HIon())} + \texttt{?kill.DeadBac()}\}_{\mu_1}$$
$$\mid \{\texttt{mov.HIon()} + \texttt{!release.HIon()}\}_{\mu_2}$$
$$\mid \{\texttt{?release.MolF()}\}_{\mu_3}$$

The process has movement steps and reactions available. We first reduce all move steps. We start with `mov.BacF()`. Assume an affine map μ_1' such that: 1) $\mu_1' = \texttt{translate(stepBF}, \mu_1, r)$, for some random number r; 2) the translated shape of the free bacterium `shapeBF` is within the movement space for free bacteria `resBF`: $\mu_1'(\texttt{shapeBF}) \subseteq \texttt{resBF}$, and 3) the new shape of the free bacterium does not overlap with the other shapes in space: $\texttt{trans}(\{\texttt{BacF()}\}_{\mu_1'}) \cap \texttt{trans}(\{\texttt{HIon()}\}_{\mu_2} \mid \{\texttt{MolB()}\}_{\mu_3}) = \emptyset$. If there is no such μ_1', the configuration remains unchanged, and the next movement step can be reduced.

Using the non-stochastic reduction rules, NR.MOVE and NR.PAR, we have:

$$\{\text{mov.BacF()} + \text{!bind.BacB()} + \text{delay@1.5.(BacF()|BacF())} + \text{delay@0.5.(BacF()|HIon())}$$
$$+ \text{?kill.DeadBac()}\}_{\mu_1} \mid \{\text{mov.HIon()} + \text{!release.HIon()}\}_{\mu_2} \mid \{\text{?release.MolF()}\}_{\mu_3}$$
$$\rightarrow \{\text{BacF()}\}_{\mu_1'} \mid \{\text{mov.HIon()} + \text{!release.HIon()}\}_{\mu_2} \mid \{\text{?release.MolF()}\}_{\mu_3}$$

We next reduce `mov.HIon()` using NR.MOVE and NR.PAR:

$$\{\text{BacF()}\}_{\mu_1'} \mid \{\text{mov.HIon()} + \text{!release.HIon()}\}_{\mu_2} \mid \{\text{?release.MolF()}\}_{\mu_3}$$
$$\rightarrow \{\text{BacF()}\}_{\mu_1'} \mid \{\text{HIon()}\}_{\mu_2'} \mid \{\text{?release.MolF()}\}_{\mu_3}$$

Unfolding $\{\text{BacF()}\}_{\mu_1'}$ and $\{\text{HIon()}\}_{\mu_2'}$ for the sake of clarity:

$$\{\text{mov.BacF()} + \text{!bind.BacB()} + \text{delay@1.5.(BacF()|BacF())} + \text{delay@0.5.(BacF()|HIon())}$$
$$+ \text{?kill.DeadBac()}\}_{\mu_1'} \mid \{\text{mov.HIon()} + \text{!release.HIon()}\}_{\mu_2'} \mid \{\text{?release.MolF()}\}_{\mu_3}$$

There are three available reactions, one communication on channel `release` and two delays. The next reaction will be determined using the corresponding reaction rates. Assume that the next reaction is the communication between `HIon()` and `MolB()` through channel `release`. Using the stochastic rules SR.COM and SR.PAR, if the distance between the processes is within `release`'s reaction radius $(\text{dis}(\mu_2', \mu_3) \leq 2.0)$ then the last process reduces to:

$$\{\text{mov.BacF()} + \text{!bind.BacB()} + \text{delay@1.5.(BacF()|BacF())} + \text{delay@0.5.(BacF()|HIon())}$$
$$+ \text{?kill.DeadBac()}\}_{\mu_1'} \mid \{\text{HIon()}\}_{\mu_2'} \mid \{\text{MolF()}\}_{\mu_3}$$

Otherwise, the configuration remains unchanged. The next reduction is a movement step, where we reduce all available `mov` prefixes. This includes reducing `mov.BacF()`, `mov.HIon()`, and `mov.MolF()`, and subsequently interleaving stochastic (reaction) and non-stochastic (movement) reductions.

6 Translation from BioScape to 3π

We define a translation from BioScape into a variant of Cardelli-Gardner's 3π with definitions instead of replication, and prove that it is sound with respect to the operational semantics (Thm. 6.1). Since 3π does not include stochasticity, we translate only the spatial aspect of BioScape into 3π.

In an attempt to make this paper self contained, we include in this section material from *Processes in Space* [7]. "3π is a proper extension of π-calculus with by-value communication of geometric data Δ, data comparisons $\Delta =_\sigma \Delta.P$, and frame shifting $\mu[P]$. The syntax of 3π is shown in Table 1. Each data term and value has a sort $\sigma \in \{\mathbf{c}, \mathbf{a}, \mathbf{p}, \mathbf{v}, \mathbf{m}\}$, denoting channels, scalars, points, vectors, and maps respectively. A geometric data can be a value or a variable, or a function on val-

$$\Delta ::= x_c \mid \ldots \mid \mu[\Delta]$$
$$\pi ::= ?_\sigma x(x') \mid !_\sigma x(\Delta) \mid \Delta =_\sigma \Delta$$
$$P ::= 0 \mid \pi.P \mid P + P' \mid P|P' \mid (\nu x)P \mid \mu[P]$$
$$D ::= \emptyset \mid D, X(x) = P$$

(Red Comm) $\Delta_{\mathcal{A}} \mapsto \epsilon$ implies $!_\sigma x(\Delta).P + P' \mid$
$?_\sigma x(y).Q + Q'_{\mathcal{A}} \rightarrow P|Q\{y/\epsilon\}$
(Red Cmp) $\Delta_{\mathcal{A}} \curlyvee \Delta'$ implies $\Delta =_\sigma \Delta'.P_{\mathcal{A}} \rightarrow P$
(Red Par) $P_{\mathcal{A}} \rightarrow Q$ implies $P \mid R_{\mathcal{A}} \rightarrow Q \mid R$
(Red Res) $P_{\mathcal{A}} \rightarrow Q$ implies $(\nu x)P_{\mathcal{A}} \rightarrow (\nu x)Q$
(Red $\equiv$) $P' \equiv P, \ P_{\mathcal{A}} \rightarrow Q, \ Q \equiv Q'$
 imply $P'_{\mathcal{A}} \rightarrow Q'$

Table 1 3π syntax and reduction

ues and variables, or a frame shift. An action term π can be an input $?_\sigma x(x')$, an output $!_\sigma x(\Delta)$, or a data comparison $\Delta =_\sigma \Delta$. The input and output actions are

analogous to π-calculus actions, while the data comparison evaluates to P if Δ and Δ' evaluate to the same value. Actions are restricted by sorting constraints: channels must have sort $\mathbf{c}$; x' in input must have sort σ; Δ in output must have sort σ; Δ and Δ' in data comparison must have sort σ.

Process terms are the standard π-calculus terms, plus the frame shift process $\mu[P]$: it means running the process P in the global frame $\mathcal{A}$ shifted by the affine map obtained by evaluating the map μ.

The reduction relation $_\mathcal{A}\mapsto$, which relates two processes relative to the global frame $\mathcal{A}$, appears in Table 1.

Reduction rules are the rule of a by-value π-calculus with data terms Δ, but `Red Comm` and `Red Cmp` rules depend on an evaluation relation $_\mathcal{A}\mapsto$, that evaluates a data Δ to value ϵ in a global frame $\mathcal{A}$. Data comparison requires the data evaluation $\Delta_\mathcal{A}\curlyvee\Delta'$, meaning there is a data value ϵ such that $\Delta_\mathcal{A}\mapsto\epsilon$ and $\Delta'_\mathcal{A}\mapsto\epsilon$."

We define three functions $\|-\|^{\mathtt{SPA}}$, $\|-\|^{\mathtt{PRO}}$ and $\|-\|^{\mathtt{DEF}}$ to translate spatial configurations, processes and definitions, respectively. In $\|A\|^{\mathtt{SPA}}_{X;E;D;\mathbf{r}}$, A is a spatial configuration, X is a triple containing the movement space, shape and step of the current definition, E is a set of channel declarations, D is a set of definitions, and r is a real number. We will abuse the notation and consider E and D sets or lists. Similarly for $\|P\|^{\mathtt{PRO}}_{X;E;D;\mathbf{r}}$ and $\|D\|^{\mathtt{DEF}}_{X;E;D';\mathbf{r}}$.

[**Process Translation**]

$$\|(\nu x@\mathbf{r},\mathbf{rad}).P\|^{\mathtt{PRO}}_{X;E;D;\mathbf{r}} = \|P\|^{\mathtt{PRO}}_{X;<x@r,rad>,E;D;\mathbf{r}} \qquad \text{P.Res}$$

$$\|P\mid Q\|^{\mathtt{PRO}}_{X;E;D;\mathbf{r}} = \|P\|^{\mathtt{PRO}}_{X;E;D;\mathbf{r}} \mid \|Q\|^{\mathtt{PRO}}_{X;E;D;\mathbf{r}} \qquad \text{P.Par}$$

$$\|X(y)\|^{\mathtt{PRO}}_{X;E;<X(x)@\xi,\sigma,\omega=M>,D;\mathbf{r}} = X(y) \qquad \text{P.Def}$$

$$\|\mathbf{0}\|^{\mathtt{PRO}}_{X;E;D;\mathbf{r}} = \mathbf{0} \qquad \text{P.Nil}$$

$$\|\pi.P+M\|^{\mathtt{PRO}}_{X;E;D;\mathbf{r}} = \{\|\pi.P\|^{\mathtt{PRO}}_{X;E;D;\mathbf{r}}\}\cup\|M\|^{\mathtt{PRO}}_{X;E;D;\mathbf{r}} \qquad \text{P.Cho}$$

$$\|\mathtt{delay}@\mathbf{r}.P\|^{\mathtt{PRO}}_{X;E;D;\mathbf{r}} = (\nu m)(!_\mathbf{c}m()\mid?_\mathbf{c}m().\|P\|^{\mathtt{PRO}}_{X;E;D;\mathbf{r}}) \quad m\notin fn(P) \qquad \text{P.Del}$$

$$\|!x(y).P\|^{\mathtt{PRO}}_{X;<x@r,rad>,E;D;\mathbf{r}} = ?_\mathbf{c}c_x(ok).?_\mathbf{c}ok(pos).?_\mathbf{p}pos(p).$$
$$abs(p-\maltese)\le rad =_\mathbf{a} 1.!_\mathbf{c}ok(y).\|P\|^{\mathtt{PRO}}_{X;<x@r,rad>,E;D;\mathbf{r}}$$
$$c_x,ok,pos,p\notin fn(P) \qquad \text{P.Out}$$

$$\|?x(z).Q\|^{\mathtt{PRO}}_{X;<x@r,rad>,E;D;\mathbf{r}} = (\nu ok,pos)(!_\mathbf{c}c_x(ok).!_\mathbf{c}ok(pos).$$
$$!_\mathbf{p}pos(\maltese).?_\mathbf{c}ok(z).\|Q\|^{\mathtt{PRO}}_{X;<x@r,rad>,E;D;\mathbf{r}}) \quad c_x\notin fn(Q) \qquad \text{P.In}$$

$$\|\mathtt{mov}.P\|^{\mathtt{PRO}}_{<\xi,\sigma,\omega>;E;D;\mathbf{r}} = into(\mathbf{T}(\uparrow(\mathbf{r},\omega))[\maltese],\sigma,\xi) =_\mathbf{a} 1.$$
$$\mathbf{T}(\uparrow(\mathbf{r},\omega))[\|P\|^{\mathtt{PRO}}_{<\xi,\sigma,\omega>;E;D;\mathbf{r}}] \qquad \text{P.Mov}$$

[**Spatial Configuration Translation**]

$$\|\{P\}_\mu\|^{\mathtt{SPA}}_{X;E;D;\mathbf{r}} = \mu[\|P\|^{\mathtt{PRO}}_{X;E;D;\mathbf{r}}] \qquad \text{S.Loc}$$

$$\|A\mid B\|^{\mathtt{SPA}}_{X;E;D;\mathbf{r}} = \|A\|^{\mathtt{SPA}}_{X;E;D;\mathbf{r}} \mid \|B\|^{\mathtt{SPA}}_{X;E;D;\mathbf{r}} \qquad \text{S.Par}$$

$$\|(\nu x@\mathbf{r},\mathbf{rad}).A\|^{\mathtt{SPA}}_{X;E;D;\mathbf{r}} = \|A\|^{\mathtt{SPA}}_{X;<x@r,rad>,E;D;\mathbf{r}} \qquad \text{S.Res}$$

[**Definition Translation**]

$$\|\emptyset\|^{\mathtt{DEF}}_{X;E;D;\mathbf{r}} = \emptyset \quad \text{D.Nil}$$

$$\|X(x)@\xi,\sigma,\omega=M,D'\|^{\mathtt{DEF}}_{X;E;<X(x)@\xi,\sigma,\omega=M>,D;\mathbf{r}} =$$
$$(X(x)=\|M\|^{\mathtt{PRO}}_{<\xi,\sigma,\omega>;E;<X(x)@\xi,\sigma,\omega=M>,D;\mathbf{r}}),\|D'\|^{\mathtt{DEF}}_{X;E;D;\mathbf{r}} \qquad \text{D.Def}$$

The translation of a BioScape choice process is the set of translations of its branches, because 3π does not allow reduction under its choice operator. Otherwise the reductions obtained from translating the branches would be blocked. Consequently, we extend naturally 3π's reduction and congruence to sets (see Table 2), and we also define homomorphic extensions of parallel composition, frame shift, prefix and restriction over sets. The trickiest rules are the ones for translating communication, `P.In` for input and `P.Out` for output. First, the input sends to the output two private channels, *ok* and *pos*, and subsequently its position (✠) via channel *pos*. The output calculates the distance between itself and the position sent by the input with the function $abs(p - ✠)$, i.e. the absolute value of the difference between the positions of the processes. If this distance is less than the reaction radius (`rad`), then the output sends a communication on the channel of success and performs the process P: when the input receives a communication on the *ok* channel, it performs process Q.

For the translation of movement in Rule `P.Mov`, we assume a function $into(p, sh, sp)$ that checks whether a shape s, centered at point p, is contained within space sp. $T(v)$ is a translation map with vector v, and $\uparrow(\mathbf{r}, \omega)$ is a vector of length ω and direction $\mathbf{r}$. The translation checks whether $into(T(\uparrow(\mathbf{r}, \omega))[✠], \sigma, \xi)$ corresponding to condition $\mu'(\sigma) \subseteq \xi$ in `NR.Move`. Finally, since delay is not a prefix in 3π, it is translated as a communication over a channel.

Theorem 6.1 (Soundness) *If $E \vdash_D A \to B$ or $E \vdash_D A \to_r B$ then there exists S such that $\|A\|^{SPA}_{\emptyset;E;D;r\mathcal{A}} \xrightarrow{*} S$ and $S \equiv \|B\|^{SPA}_{\emptyset;E;D;r}$ for some number r.*

In our translation, the global frame $\mathcal{A}$ is the identity. Although sound, this translation is not complete in the sense that not all reductions in the translated code arise from corresponding reductions in the source code. Achieving completeness is more complex and would require collision detection and a more refined mechanism for checking proximity.

Let S and T be sets

(i) $S \mid T = \{(P \mid Q) \mid P \in S, Q \in T\}$

(ii) $\mu[\{\emptyset\}] = \emptyset$ and
       $\mu[\{P\} \cup S] = \{\mu[P]\} \cup \mu[S]$

(iii) $\pi.S = \{\pi.P \mid P \in S\}$

(iv) $(\nu m).S = \{(\nu m).P \mid P \in S\}$

(Set.Par) $S_{\mathcal{A}} \to S'$ implies $S \mid T_{\mathcal{A}} \to S' \mid T$

(Set.Red) $P_{\mathcal{A}} \to P'$ imp. $\{P\} \cup Q_{\mathcal{A}} \to \{P'\} \cup Q$

(S.Set.Par) $S \equiv S'$ imp. $S \mid T \equiv S' \mid T$

(S.Set.Red) $P \equiv P'$ imp. $\{P\} \cup Q \equiv \{P'\} \cup Q$

Table 2 Operations, Reduction and Congruence on Sets

7 Conclusions

We define BioScape for the modeling and simulation of complex bacteria-materials interactions. BioScape builds on 3π [7] and SpacePi [15] merging an affine space geometry, reaction radius [1] and reaction boundary. This combination of features is strictly motivated by the nature of the models we are capturing, as described in the introduction: stochasticity, movement, individual process location (in contrast with homogeneously mixed reactants in a volume), interaction in proximity, and movement confinement. We formulate a reduction semantics for BioScape and

---

[1] Although this concept may be derived in 3π, it is convenient in practice to have as primitive.

demonstrate it in an example of pH-triggered drug release in the presence of bacterial infection. Furthermore, we validate BioScape with a translation into Cardelli-Gardner's 3π, and prove its soundness with respect to the operational semantics. Complete technical details can be found in the companion technical report [11].

The process calculus that most closely resembles BioScape is the Shape Calculus [3], a CCS-like timed calculus, with simulating tool BioShape [5]. These are some of the differences. BioScape is stochastic, but the Shape Calculus is not. The Shape Calculus does not allow dynamic creation of channels, but BioScape does. Agents in BioScape can be modified by affine transformations useful in modeling phenomena such as cell growth, but the Shape Calculus only allows movement specified with a velocity vector. The Shape Calculus has a time primitive for describing a delay, while BioScape has a stochastic delay. The specification of an agent in BioScape describes an area where it is allowed to be, but the Shape Calculus does not. This area is instrumental in describing biomaterials such as antibacterial surfaces and preventing bacteria from penetrating the surface while allowing antibacterial molecules to do so. On the other hand, such behavior would have to be programmed in the description of the agent in the Shape Calculus.

We develop an implementation of BioScape based on SPiM, and show preliminary simulation results in agreement with wet-lab experiments. We are currently working on a parallel simulation algorithm implemented in CUDA, and our prototype implementation can handle one million agents in 3D space.

Our long-term modeling objective is to identify combinations of substrate variables that most significantly inhibit bacterial colonization and promote tissue integration.

Acknowledgement

References

[1] C. R. Arciola. New concepts and new weapons in implant infections. *International Journal of Artificial Organs*, 32(9):533–536, 2009.

[2] Y. Bao, A. B. Compagnoni, J. Glavy, and T. White. Computational modeling for the activation cycle of G-proteins by G-protein-coupled receptors. In *Proceedings Fourth Workshop on Membrane Computing and Biologically Inspired Process Calculi 2010. (MeCBIC)*, number 40 in Electronic Proceedings in Theoretical Computer Science, pages 39–53, 2010.

[3] E. Bartocci, F. Corradini, M. R. D. Berardini, E. Merelli, and L. Tesei. Shape calculus. a spatial mobile calculus for 3d shapes. *Sci. Ann. Comp. Sci.*, 20:1–31, 2010.

[4] A. T. Bittig and A. M. Uhrmacher. Spatial modeling in cell biology at multiple levels. In *Winter Simulation Conference*, pages 608–619, 2010.

[5] F. Buti, D. Cacciagrano, F. Corradini, E. Merelli, and L. Tesei. Bioshape: a spatial shape-based scale-independent simulation environment for biological systems. *Procedia Computer Science*, 1(1):827 – 835, 2010.

[6] L. Cardelli, E. Caron, P. Gardner, O. Kahramanogullari, and A. Phillips. A process model of rho gtp-binding proteins. *Theor. Comput. Sci.*, 410(33-34):3166–3185, 2009.

[7] L. Cardelli and P. Gardner. Processes in space. In *CiE*, pages 78–87, 2010.

[8] F. Castiglione and M. Bernaschi. C-immsim: playing with the immune response. In *Proceedings of the Sixteenth International Symposium on Mathematical Theory of Networks and Systems (MTNS2004)*, 2004.

[9] R. G. Chapman, E. Ostuni, M. Liang, G. Meluleni, E. Kim, L. Yan, G. Pier, H. Warren, and G.M. Whitesides, polymeric thin films that resist the adsorption of proteins and the adhesion of bacteria. *Langmuir*, 17(4):1225–1233, 2001.

[10] F. Ciocchetta and J. Hillston. Bio-pepa: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, 410(33-34):3065–3084, 2009.

[11] A. Compagnoni, V. Sharma, Y. Bao, P. Bidinger, L. Bioglio, E. Bonelli, M. Libera, and S. Sukhishvili. Bioscape: A modeling and simulation language for bacteria-materials interactions. Technical Report CS-1-2012, Stevens Institute of Technology, NJ, 2012.

[12] V. Danos, J. Feret, W. Fontana, R. Harmer, and J. Krivine. Rule-based modelling of cellular signalling. In L. Caires and V. T. Vasconcelos, editors, *CONCUR*, volume 4703 of *Lecture Notes in Computer Science*, pages 17–41. Springer, 2007.

[13] R. O. Darouiche. Treatment of infections associated with surgical implants. *New England Journal of Medicine*, 350(14):1422–1429, 2004.

[14] N. P. Desai, S. Hossainy, and J. Hubbell. Surface-immobilized polyethylene oxide for bacterial repellence. *Biomaterials*, 13(7):417–420, 1992.

[15] M. John, R. Ewald, and A. M. Uhrmacher. A spatial extension to the pi calculus. *Electron. Notes Theor. Comput. Sci.*, 194:133–148, January 2008.

[16] M. Kazemzadeh-Narbat, J. Kindrachuk, K. Duan, H. Jenssen, R. Hancock, and R. Wang. Antimicrobial peptides on calcium phosphate-coated titanium for the prevention of implant-associated infections. *Biomaterials*, 31(36):9519–9526, 2010.

[17] B. N. Kholodenko. Cell-signalling dynamics in time and space. *Nature Reviews Molecular Cell Biology*, 7(3):165–176, 2006.

[18] X. Khoo, G. O'Toole, S. Nair, B. Snyder, D. Kenan, and M. Grinstaff. Staphylococcus aureus resistance on titanium coated with multivalent pegylated-peptides. *Biomaterials*, 31(35):9285–9292, 2010.

[19] P. Kingshott, J. Wei, D. Bagge-Ravn, N. Gadegaard, and L. Gram. Covalent attachment of poly(ethylene glycol) to surfaces, critical for reducing bacterial adhesion. *Langmuir*, 19(17), 2003.

[20] M. C. Lawson, C. Bowman, and K. Anseth. Vancomycin derivative photopolymerized to titanium kills s. epidermidis. *Clinical Orthopaedics and Related Research*, 461:96–105, 2007.

[21] C. Macal and M. North. Tutorial on agent-based modelling and simulation. *Journal of Simulation*, 4(3):151–162, 2010.

[22] J. S. Moskowitz, M. Blaisse, R. Samuel, H. Hsu, M. Harris, S. Martin, J. Lee, M. Spector, and P. Hammond. The effectiveness of the controlled release of gentamicin from polyelectrolyte multilayers in the treatment of staphylococcus aureus infection in a rabbit bone model. *Biomaterials*, 31(23):6019–6030, 2010.

[23] V. Mugathan, A. Phillips, and M. Vigliotti. Bam: Bioambient machine. In *Application of Concurrency to System Design*, pages 45–49. IEEE Computer Society, June 2008.

[24] N. L. Novère and T. S. Shimuzu. Stochsim: modelling of stochastic biomolecular processes. *BioInformatics*, 17(6):575–576, 2001.

[25] S. Pavlukhina, Y. Lu, A. Patimetha, M. Libera, and S. Sukhishvili. Polymer multilayers with ph-triggered release of antimicrobial agents. *Langmuir*, 11(12):3448–3456, 2007.

[26] M. Peleg, D. Rubin, and R. B. Altman. Using Petri Net Tools to Study Properties and Dynamics of Biological Systems. *Journal of the American Medical Informatics Association*, 12(2):181–199, 2005.

[27] A. Phillips and L. Cardelli. Efficient, correct simulation of biological processes in the stochastic pi-calculus. In *Computational Methods in Systems Biology*, volume 4695 of *LNCS*, pages 184–199. Springer, September 2007.

[28] N. F. Polys, D. A. Bowman, C. North, R. Laubenbacher, and K. Duca. Pathsim visualizer: an information-rich virtual environment framework for systems biology. In *Web3D 04: Proceedings of the ninth international conference on 3D Web technology*, pages 7–14, New York, NY, USA, 2004. ACM, ACM.

[29] C. Priami. Stochastic pi-calculus. *Comput. J.*, 38(7):578–589, 1995.

[30] C. Priami, A. Regev, E. Y. Shapiro, and W. Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Inf. Process. Lett.*, 80(1):25–31, 2001.

[31] A. Radu, J. Vrouwenvelder, M. van Loosdrecht, and C. Picioreanu. Modeling the effect of biofilm formation on reverse osmosis performance: Flux, feed channel pressure drop and solute passage. *Journal of Membrane Science*, 365(1-2):1–15, 2010.

[32] N. Rapin, O. Lund, and F. Castiglione. Immune system simulation online. *Bioinformatics*, 2011.

[33] A. Roosjen, J. D. Vries, H. V. D. Mei, W. Norde, and H. Busscher. Stability and effectiveness against bacterial adhesion of poly(ethylene oxide) coatings in biological fluids. *Journal of Biomedical Materials Research - Part B Applied Biomaterials*, 20(25):347–354, 2005.

[34] A. Stefanek, M. Vigliotti, and J. T. Bradley. Spatial extension of stochastic pi calculus. In *8th Workshop on Process Algebra and Stochastically Timed Activities*, pages 109–117, 2009.

[35] A. Trampuz and A. Widmer. Infections associated with orthopedic implants. *Current Opinion in Infectious Diseases*, 19(4):349–356, 2006.

[36] D. S. Wishart, R. Yang, D. Arndt, P. Tang, and J. Cruz. Dynamic cellular automata: an alternative approach to cellular simulation. *In Silico Biology*, 5(2):139–161, 2005.

[37] Y. Wu, J. Zitelli, K. TenHuisen, X. Yu, and M. Libera. Differential response of staphylococci and osteoblasts to varying titanium surface roughness. *BioMaterials*, 32(4):951–960, 2011.

[38] J. B. Xavier, M. K. de Kreuk, C. Picioreanu, and M. C. M. van Loosdrecht. Multi-scale individual-based model of microbial and bioconversion dynamics in aerobic granular sludge. *Environmental Science and Technology*, 41(18):6410–6417, 2007.