

RESEARCH ARTICLE

Open Access



PROKARYO: an illustrative and interactive computational model of the lactose operon in the bacterium *Escherichia coli*

Afshin Esmaili^{1†}, Timothy Davison^{1†}, Andrew Wu¹, Joenel Alcantara³ and Christian Jacob^{1,2*}

Abstract

Background: We are creating software for agent-based simulation and visualization of bio-molecular processes in bacterial and eukaryotic cells. As a first example, we have built a 3-dimensional, interactive computer model of an *Escherichia coli* bacterium and its associated biomolecular processes. Our illustrative model focuses on the gene regulatory processes that control the expression of genes involved in the lactose operon. *Prokaryo*, our agent-based cell simulator, incorporates cellular structures, such as plasma membranes and cytoplasm, as well as elements of the molecular machinery, including RNA polymerase, messenger RNA, lactose permease, and ribosomes.

Results: The dynamics of cellular ‘agents’ are defined by their rules of interaction, implemented as finite state machines. The agents are embedded within a 3-dimensional virtual environment with simulated physical and electrochemical properties. The hybrid model is driven by a combination of (1) mathematical equations (DEQs) to capture higher-scale phenomena and (2) agent-based rules to implement localized interactions among a small number of molecular elements. Consequently, our model is able to capture phenomena across multiple spatial scales, from changing concentration gradients to one-on-one molecular interactions. We use the classic gene regulatory mechanism of the lactose operon to demonstrate our model’s resolution, visual presentation, and real-time interactivity. Our agent-based model expands on a sophisticated mathematical *E. coli* metabolism model, through which we highlight our model’s scientific validity.

Conclusion: We believe that through illustration and interactive exploratory learning a model system like *Prokaryo* can enhance the general understanding and perception of biomolecular processes. Our agent-DEQ hybrid modeling approach can also be of value to conceptualize, illustrate, and—eventually—validate cell experiments in the wet lab.

Keywords: Computational modeling, Mathematical model, Agent-based model, *E. coli* simulation, Lactose operon model

Background

A bacterial cell, as elementary as it might be from a biological perspective, is a good model organism to study biological complexity. Illustrations and animations are powerful ways to explore and describe complex systems. David Goodsell’s book “The Machinery of Life”, in which *E. coli*

bacteria and biomolecules play a prominent role, is an excellent example of how to communicate scientific concepts through textual descriptions in combination with illustrative drawings of cellular structures across a range of scales [1]. Through such illustrations, molecular and cellular structures become tangible and attain meaning within their specific metabolic contexts. This promotes a deeper scientific understanding of the systems at hand.

Motivated by Goodsell’s visuals, we have taken his highly detailed, illustrative “snapshots” one step further: we bring the biomolecular interactions of a bacterial cell alive as 3-dimensional computer simulations (Fig. 1). The output from our *Prokaryo* model is similar to the animations generated through Harvard University’s BioVisions

*Correspondence: cjacob@ucalgary.ca

†Equal contributors

¹Department of Computer Science, Faculty of Science, University of Calgary, 2500 University Drive NW, T2N 1N4 Calgary, Canada

²Department of Biochemistry and Molecular Biology, Cumming School of Medicine, University of Calgary, 3330 Hospital Drive NW, T2N 4N1 Calgary, Canada

Full list of author information is available at the end of the article

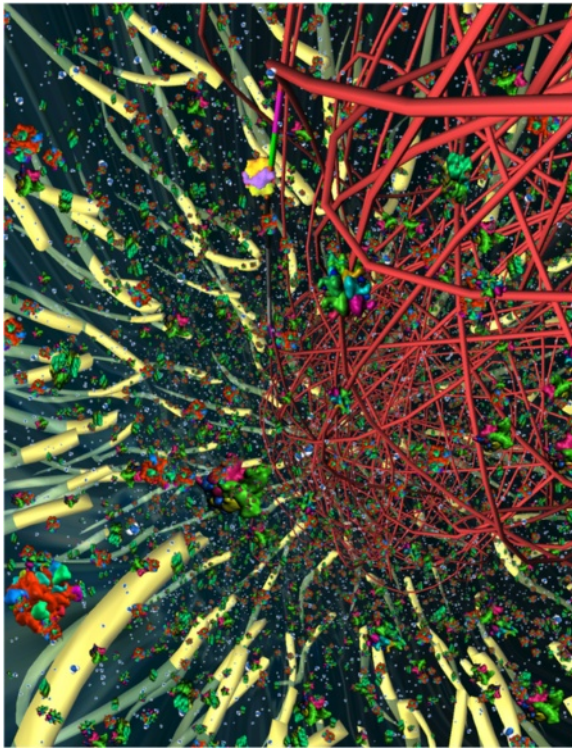


Fig. 1 The *Prokaryo* cytoplasm. A snapshot from an interactive simulation with DNA structure, water, ribosomes, RNA polymerases, β -galactosidase and lactose. In this scene over 70,000 particles are rendered in realtime. Compare Fig. 11 for protein shapes and colour representations

project [2]. In order to visualize the inner workings of a eukaryotic cell, Harvard BioVisions produced an eight-minute animation entitled “The Inner Life of a Cell” [3]. Unlike Goodsell’s static illustrations, BioVisions enhances the understanding of structural and cellular biology by providing movement, flow, and a sense of real dynamics. However, an important element is missing in these animations: a way to interact with and explore the models in real time.

Interactivity enables inquiry-based investigation through self-directed exploration, which is a powerful and effective way of learning [4]. This is especially true for comprehending complex system dynamics. Imagine being immersed in a bacterial cell, cruising along the cell surface, slipping through the membrane, diving into the cytoplasm, and exploring the dynamic worlds inside a cell — all under your own navigational control.

In this paper we show a first step in this direction with *Prokaryo*, an illustrative, interactive 3D model with integrated simulations of biomolecular processes inside an *E. coli* cell. Our approach combines a sophisticated mathematical model [5] and an agent-based approach to simulate regulatory processes of the lactose operon.

Biological background: the bacterium *Escherichia coli*

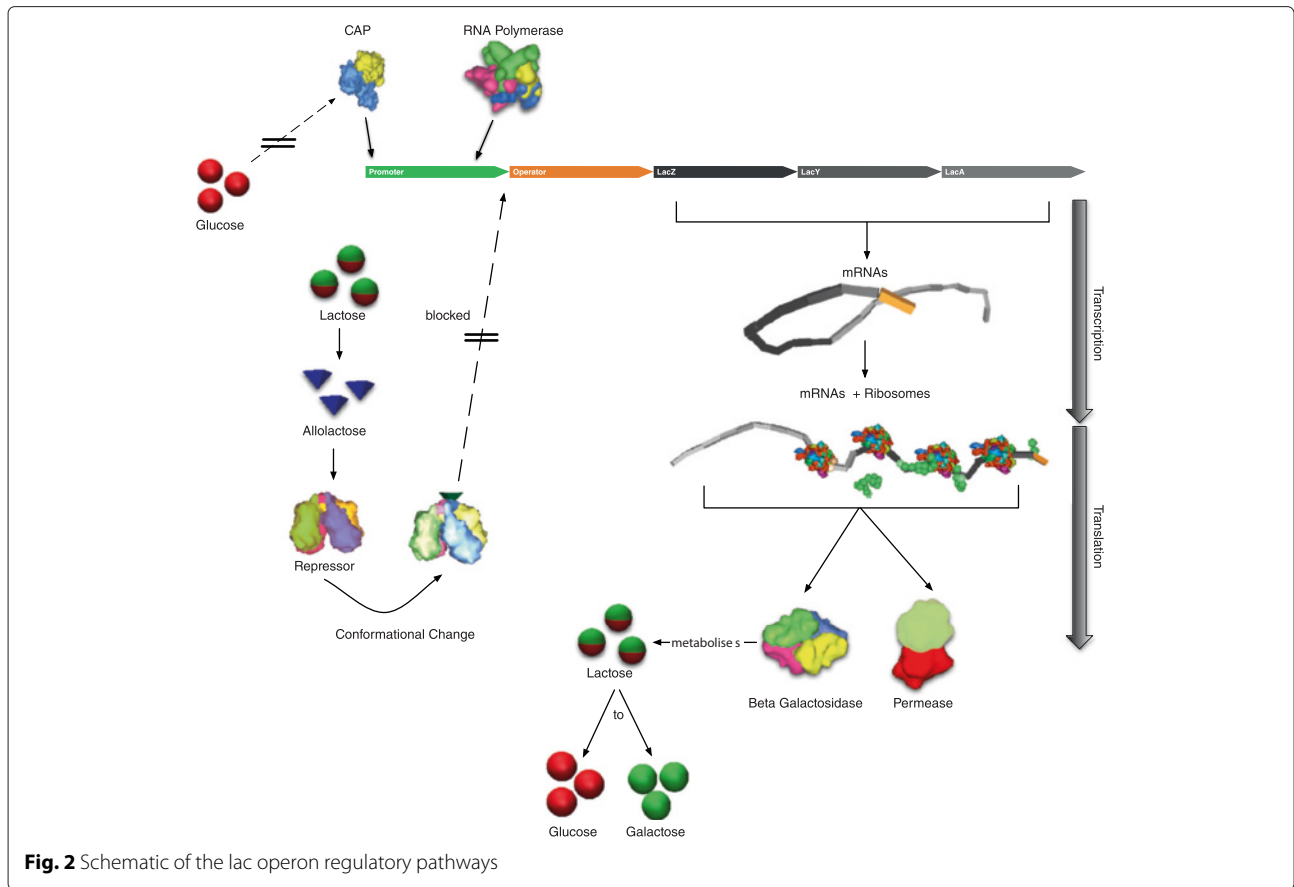
Billions of bacteria live inside our intestinal tracts with an estimated 100 billion billion bacteria on Earth [6]. Due to its ease of growth and versatility as an organism, *Escherichia coli* (*E. coli*) has been the centre of many biological discoveries. With a genome that encodes for 4300 proteins, *E. coli* is one of the first organisms to have its DNA sequenced, which provides the basis for understanding the genetic programs of a bacterial cell [1].

Surprisingly though, exact details of how a biological cell works—to a large extent—still remain a mystery. This is even the case for comparatively simple bacterial (prokaryotic) cells. Eukaryotic cells are even more complex, and consequently even harder to model [7]. A good starting point for understanding the molecular dynamics inside a bacterium is to investigate how its small building blocks (molecules) interact with other building blocks and structural elements (cytoplasm, periplasm) within the cell. For instance, proteins acting as repressors regulate gene expressions, which further trigger a cascade of events. Understanding the regulation processes is crucial for identifying cellular responses to internal and external signals. As a cell reacts to signals by switching different genes on and off, different proteins are manufactured in response [8]. Given the prominent role of gene regulation in a cell’s life cycle, we have chosen a classical, well-studied gene regulation mechanism inside *E. coli* to be modeled and simulated as part of this work: the lactose operon switch (Fig. 2).

In the following section, we explain the lactose operon switching mechanisms by using illustrations taken from our *Prokaryo* model, which we will discuss in more detail later.

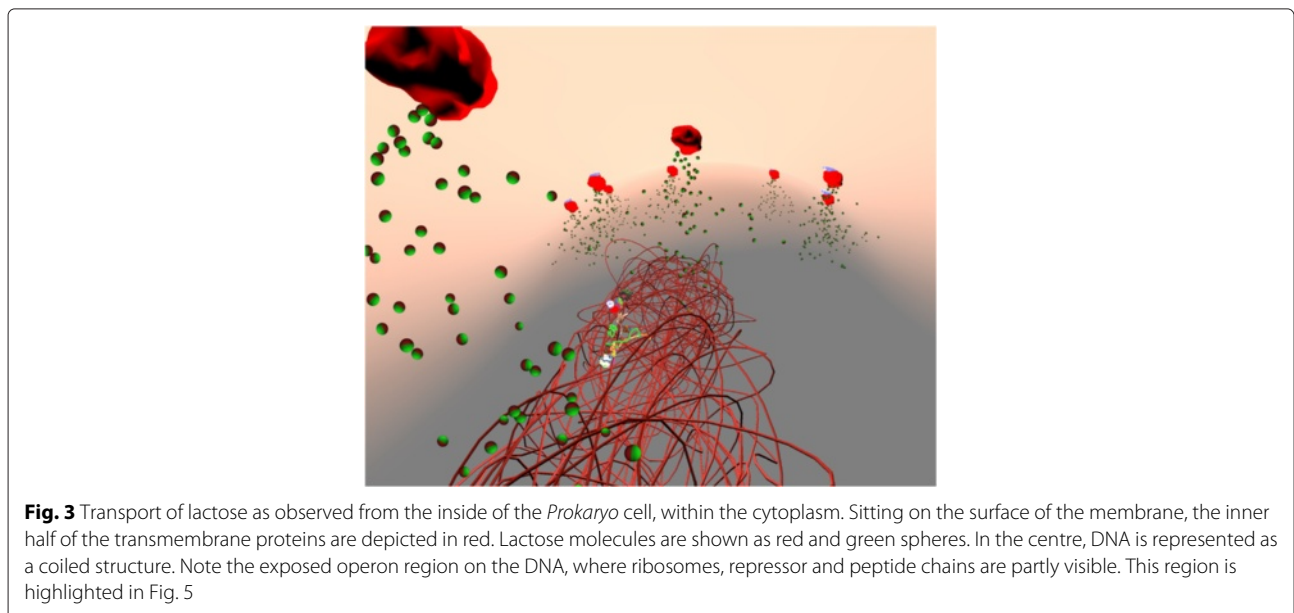
The lactose operon and the *Prokaryo* virtual cell

Half a century ago, Jacob and Monod laid the foundations of molecular biology by illustrating an example mechanism for gene regulation: the lactose operon [9]. *E. coli* thrives in a lactose-rich environment. After lactose is transported through the periplasm into the cytoplasm (Fig. 3), lactose needs to be dissociated into glucose and galactose (Fig. 4). Glucose is one of the sources of energy for the bacterium. Consequently, a protease, in this case β -galactosidase, needs to be expressed to perform the cutting. A more complete schematic view of the protein interactions involved in the lac operon is presented in Fig. 2. The three structural genes of the lac operon — lacZ, lacY and lacA (Fig. 5) — code for β -galactosidase, lactose permease and thiogalactoside transacetylase, respectively. β -galactosidase metabolizes lactose into glucose and galactose. Permease is a transmembrane protein necessary for lactose uptake. We ignore production of



thiogalactoside transacetylase due to its lack of participation in the actual lac operon regulation [10]. The lac repressor acts as a negative regulator, which prevents RNA polymerase from transcribing when lactose levels are low. In the presence of allolactose (a bi-product of lactose

metabolism) a complex forms between allolactose and the repressor. This allolactose-repressor complex causes a conformational change in the repressor and inactivates it. As a consequence, the allolactose-repressor complex is unable to bind to the operator region of the lactose



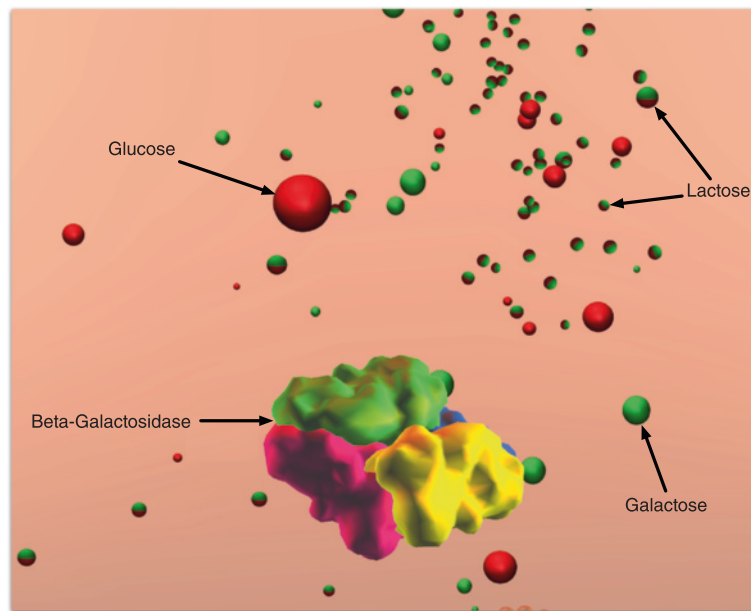


Fig. 4 β -galactosidase metabolism. Upon collision, β -galactosidase metabolizes lactose molecules into glucose and galactose. The lactose molecules are represented as half red and half green spheres. All red spheres depict glucose molecules; green spheres represent galactose

promoter, thus leaving the operator site accessible for RNA polymerase (Fig. 6).

The lac operon consists of three operators O_1 , O_2 and O_3 . A repressor bound to O_1 inhibits transcription initiation at a higher rate compared to a repressor bound to O_2 and O_3 which has almost no inhibitory effect [11, 12]. In this paper, we consider repressor binding to operator O_1 only. This is obviously an oversimplification of the actual competitive binding events. We will demonstrate, however, that through our modular, agent-based approach one can add competitive binding into our simulation. We have captured and described competitive binding in

another agent-based model of the λ -switch gene regulatory mechanism [13].

The second regulatory mechanism (positive regulation) in the lac operon is controlled by glucose, which is *E. coli*'s preferred carbon and energy source. As the concentration of extracellular glucose decreases, the intracellular production of cyclic AMP (cAMP) increases. cAMP binds to cAMP receptor protein (CRP) to form the cAMP-CRP complex, also known as catabolite activator protein (CAP). The CAP complex binds just upstream of the lactose promoter and assists (through positive regulation) the RNA polymerase in transcribing the lac

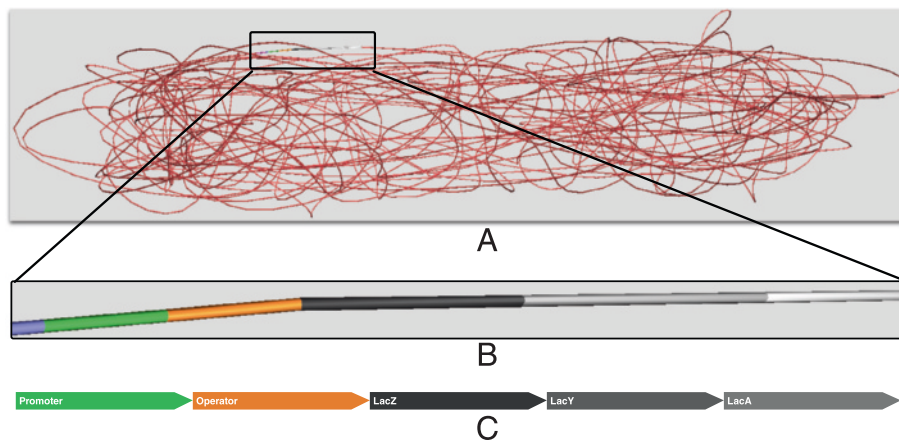


Fig. 5 Colour coded representation of the genetic sections that comprise the lactose operon. **a** Operon location relative to the DNA structure; **b** Colour coding of gene segments used in the simulation; **c** corresponding labels of gene segments

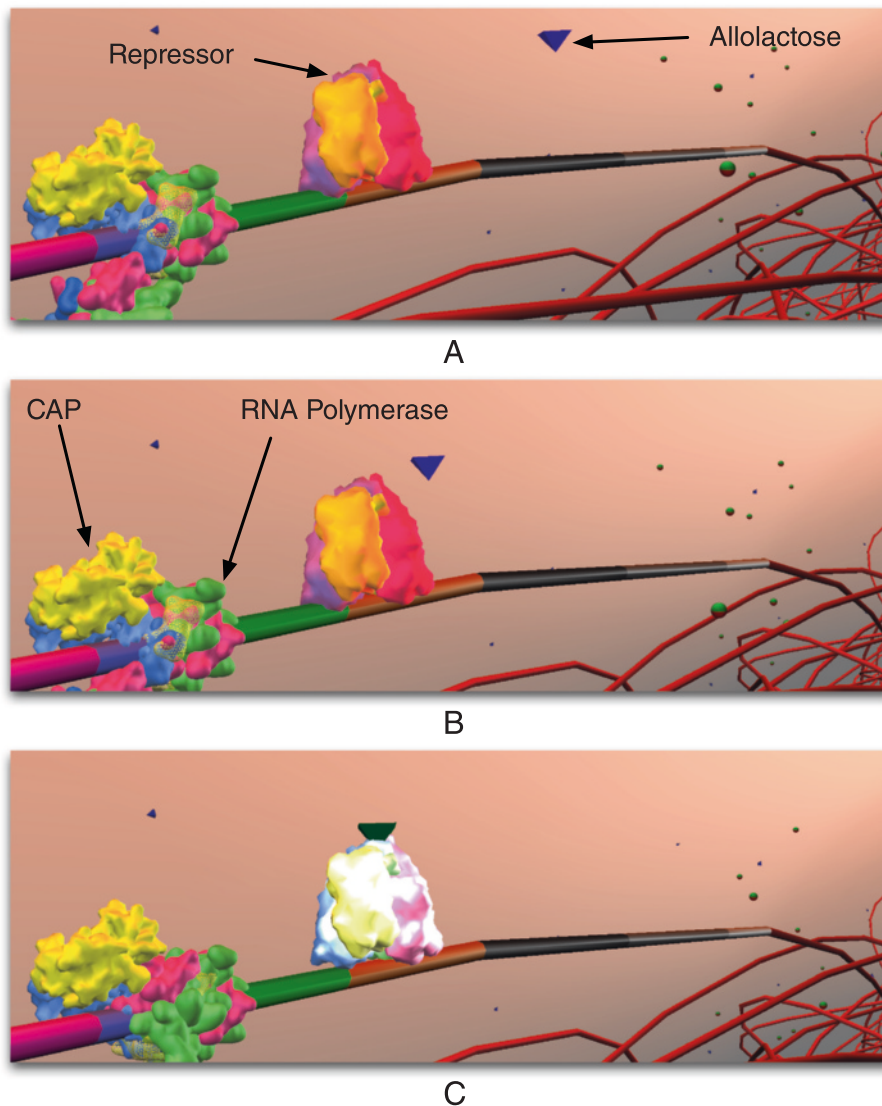


Fig. 6 Repressor Inactivation. **a** and **b** RNA polymerase and CAP are blocked by the repressor. Allolactose approaches repressor. **c** Allolactose has docked onto repressor, causing its conformational change (visualized by a colour change of the repressor protein). Subsequently, repressor will undock from DNA, initiating transcription (Fig. 7)

operon (Fig. 6b). The lac genes are regulated by CAP; this increases the affinity of RNA polymerase to transcribe the operon. Thus, the lac operon is subject to negative (repressor) and positive (CAP) regulation. At low glucose and high lactose levels, CAP is bound to the promoter, and the repressor is inactivated in the presence of allolactose.

RNA polymerase binds to the promoter and initiates transcription to produce messenger RNA (mRNA) (Fig. 7). Ribosomes translate mRNA into peptide chains, which fold into β -galactosidase and permease. For folding of peptide chains, we use a simplified folding heuristic where amino acid chains are simulated with peptide bonds at random locations (Fig. 8). Subjected to simulated physics [14], the illusion of

protein folding is created. Due to *Prokaryo's* modular, component-based system architecture, the program module that encapsulates protein folding can be enhanced at a later time, once we have a more comprehensive understanding of the molecular interaction processes involved.

Methods

Computational modeling and analysis provide insights into complex systems and can be used in education and research. Two common approaches for developing complex system abstractions are Differential Equation (DE) models [8] and multi-agent (MA) models [15]. Both modeling paradigms have their own strengths and weaknesses.

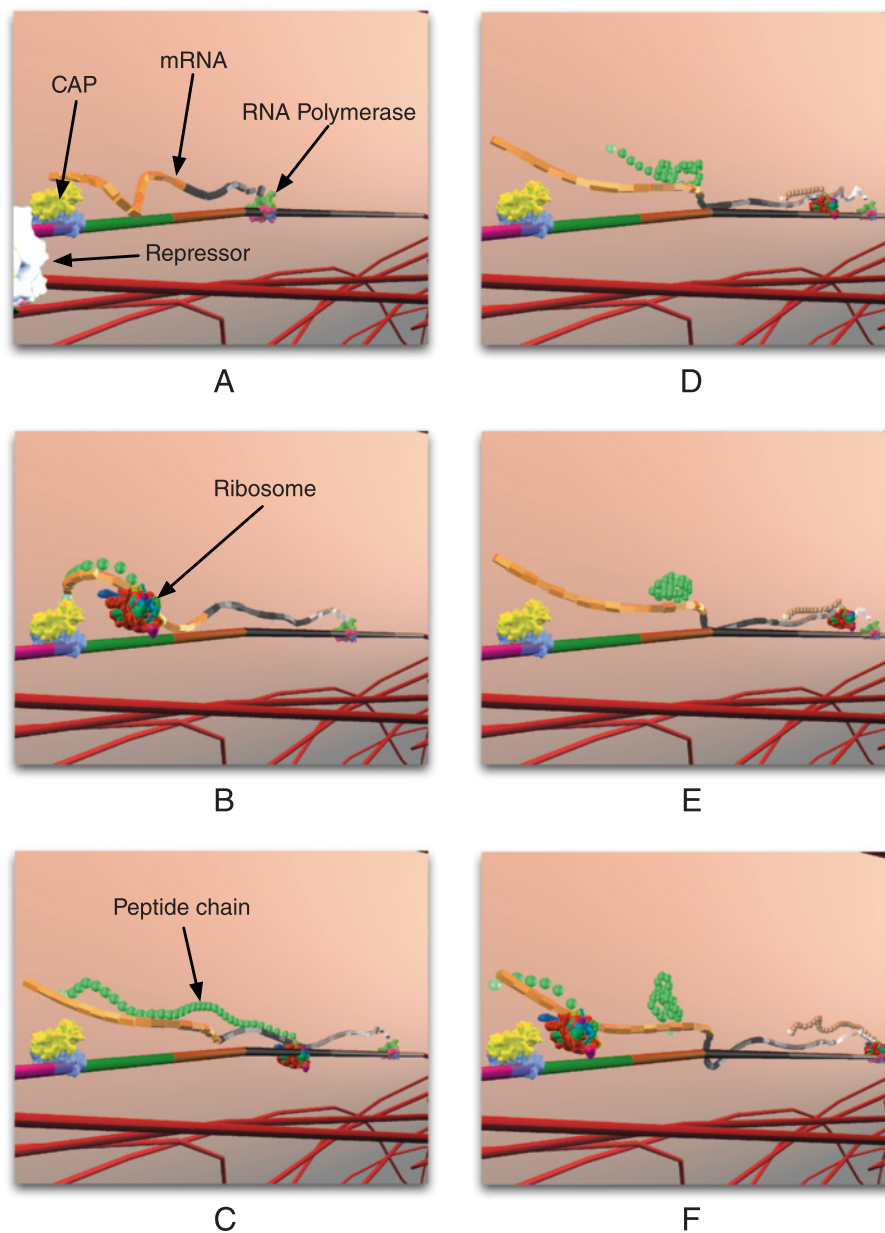


Fig. 7 Transcription and Translation. **a** Polymerase initiates transcription and generates mRNA; **(b and c)** Ribosomes initiate translation on each mRNA; **(d)** Translation of mRNA generates peptide chains, which **(e)** fold and get converted to a protein structure (compare Fig. 8); **(f)** Multiple ribosomes translate mRNA at the same time

The main challenge in building a bacterium model *in silico* is to capture emergent properties that none of the constituent parts alone possesses. Two general modeling approaches are explored in order to model and simulate a bacterium and to view the cellular processes from a holistic perspective. First, we use parameterized systems of DEs that describe the lac operon dynamics (Table 1). Second, a hybrid agent-based approach models the lac operon's heterogeneous entities (agents) in an evolving network of interactions (Figs. 6 and 7).

Other whole cell in-silico simulation systems have been developed, most notably *Virtual Cell* [16] and *Smoldyn* [17]. *Virtual Cell* converts mathematical descriptions of reaction networks into differential equations handed to numerical solvers. The output is software code that can be used for further simulation analysis. *Virtual Cell* offers graphical interfaces to define reactions and to access shared simulations. In contrast to *Prokaryo*, no user interaction is possible during the simulations. *Smoldyn*, on the other hand, simulates molecules as individual units,

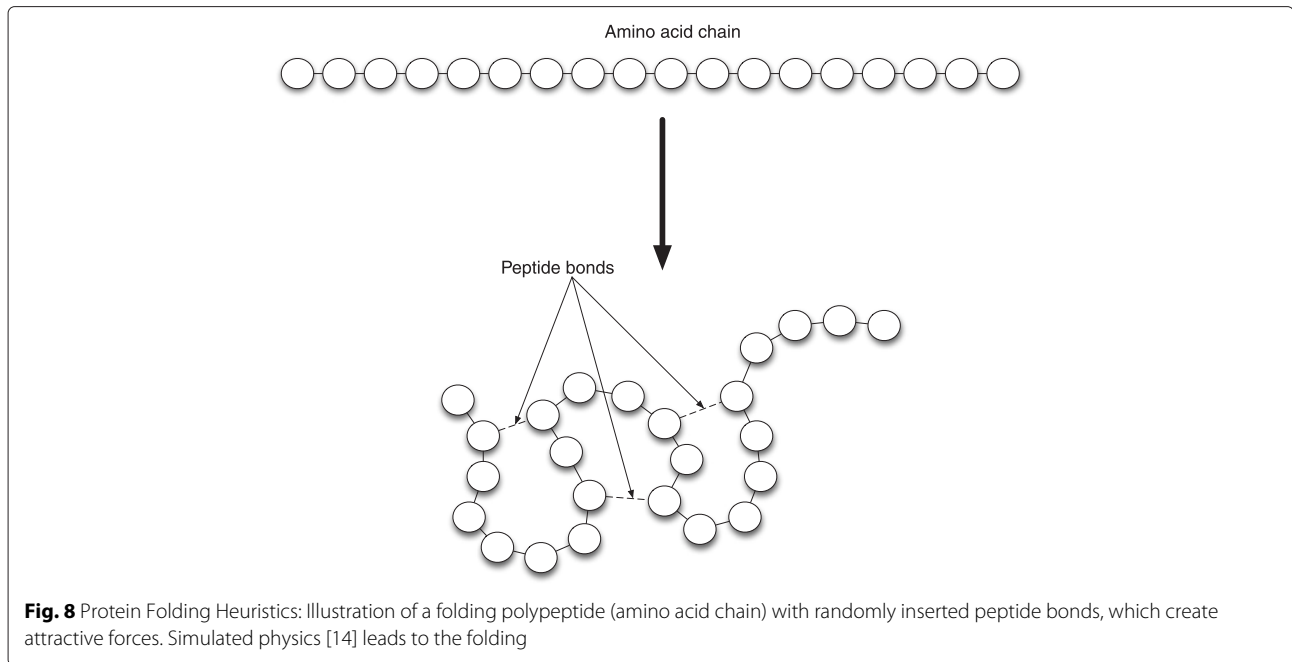


Fig. 8 Protein Folding Heuristics: Illustration of a folding polypeptide (amino acid chain) with randomly inserted peptide bonds, which create attractive forces. Simulated physics [14] leads to the folding

which diffuse and react, and captures natural stochasticity in cell scale environments. Although various forms of output are possible (data files, image sequences, movies), *Smoldyn* only offers a command line interface, and is not built for realtime interaction during simulations. In contrast to both *Virtual Cell* and *Smoldyn*, *Prokaryo*'s focus is on real-time interaction in simulated 3D cell spaces, with emphasis on illustrative and educational presentation of its simulations. *Prokaryo*'s code base can handle

general cell simulations, such as *Virtual Cell* and *Smoldyn* but is here primarily presented as a simulator to capture molecular reactions related to the lactose operon.

Table 1 Mathematical model to capture counts of intercellular mRNA (M), LacZ (E) and lactose (L) as reproduced from [27]. A summary of key symbols is listed in Table 2 with constants explained in Table 4

$$\dot{M} = Dk_M P_D P_R - \gamma_M M \tag{5}$$

$$\dot{E} = k_E M - \gamma_E E \tag{6}$$

$$\dot{L} = k_L \beta_L \beta_G Q - 2\phi_M \mathcal{M} B - \gamma_L L \tag{7}$$

$$A = L \tag{8}$$

$$Q = E \tag{9}$$

$$B = E/4 \tag{10}$$

$$P_D = \frac{p_p(1 + p_c(k_{pc} - 1))}{1 + p_p p_c(k_{pc} - 1)} \tag{11}$$

$$p_c = \frac{K_G^{th}}{K_G^{th} + G e^{th}} \tag{12}$$

$$P_R = \frac{1}{1 + \rho(A) + \frac{\xi_{122} \rho(A)}{(1 + \xi_2 \rho(A))(1 + \xi_3 \rho(A))}} \tag{13}$$

$$\rho(A) = \rho_{max} \left(\frac{K_A}{K_A + A} \right)^4 \tag{14}$$

$$\beta_L = \frac{l_e}{k_l + l_e} \tag{15}$$

$$\beta_G = 1 - \phi_G \frac{G_e}{K_G + G_e} \tag{16}$$

$$\mathcal{M} = \frac{l}{k_M + l} \tag{17}$$

Mathematical models of the lac operon

Mathematical modeling has had enormous success in disciplines such as physics, astronomy, social sciences and engineering. Biological systems, with their high level of complexity and lack of quantitative information, have been a great challenge to model. However, with recent developments of new experimental methods in generating vast amounts of data and technological advancements in the data processing power of computers, there is a renewed interest in modeling biological systems [18]. Cellular processes such as lactose metabolism and transport can be approximated by a network of chemical reactions. First examinations of the lac operon go back to nearly half a century ago when Jacob and Monod [9] as well as Novick and Wiener [19] examined the lac operon's induction mechanism as an on/off switch.

More recent approaches to modeling the lac operon dynamics have been reviewed by Santillan and Mackey [20], who present an in-depth overview of current advancements in the field of lac operon models. Following their review, we provide a short background on these modeling approaches.

One of the most detailed models of the lac operon, which relies heavily on experimental data, is the work of Wong et al. [21]. Based on a 13-dimensional DE system, their model includes (1) catabolite repression, where glucose represses synthesis of β -galactosidase and permease

by inhibiting the production of cAMP, and (2) inducer exclusion, reducing the efficiency of lactose permease to transport lactose molecules into the cell.

Vilar et al. [18] suggest a simpler model by integrating different scales into their equations. They consider the molecular, cellular and population level dynamics of the lac operon, yet ignore both catabolite repression and inducer exclusion in their model.

Yildirim & Mackey [22] modeled the lac operon dynamics through a 5-dimensional equation system. The focus of Yildirim and Mackey's work lies on the dynamics of β -galactosidase, permease, intercellular lactose and allolactose. Catabolite repression and inducer exclusion is ignored. This model considers delays in transcription and translation and their effects on the lac operon dynamics.

Santillan & Mackey [23] developed a model of both regulatory mechanisms of the lac operon – catabolite repression and inducer exclusion – using a 6-dimensional model, including all three operators acting on RNA polymerase to enhance transcription of the lac operon.

Van Hoek and Hogeweg [24, 25] constructed a population based mathematical model of the lac operon evolution. They investigated the lac operon switch response to lactose and artificial inducers via the introduction of stochasticity to simulating protein production in bursts.

Santillan et al. [5] investigated bistability of the lac operon gene regulatory system and validated their results with experimental data from Ozbudak et al. [26]. Santillan et al. developed a model of the lac operon system that indicated that bistability guarantees the metabolism of lactose only when the carbon source (glucose) is not available. Later, Santillan et al. [27] extended their work with a model that included variable growth rates in *E. coli*. Both models are DE-based using Gillespie's Tau-Leap algorithm [28, 29].

The models reviewed above are based on ordinary differential equations (ODE) with chemical kinetics formalism. ODEs are only valid, however, when the molecular counts are very large, which is actually not the case for the regulatory units in the lactose operon switch [20]. The next section describes the lac operon ODE model, based on the work of Santillan et al., that we use in our *Prokaryo* system.

The Santillan bistability model of the lac operon

Santillan et al.'s [5] model focuses on bistability of the lac operon. Tables 1 and 2 list the Santillan model equations and the functions used to describe the intermolecular dynamics. We have chosen this model as our starting point based on its minimalistic approach and its improvements over the earlier models discussed above. The Santillan model captures mRNA (M), lacZ polypeptide (E) and internal lactose (L) concentrations.

Table 2 Abbreviations of mathematical functions used in the equation set of Table 1

Lac operon mathematical function abbreviations	
G_e	Extracellular glucose
L_e	Extracellular lactose
A	Intracellular allolactose
Q	Intracellular permease
B	Intracellular β -Galactosidase
P_D	Negative effect of external glucose on the initiation rate of transcription (via catabolite repression)
P_R	Probability that the lactose promoter is not repressed
β_L	Positive effect of external lactose on its uptake rate
β_G	Negative effect of external glucose on lactose uptake (inducer exclusion)

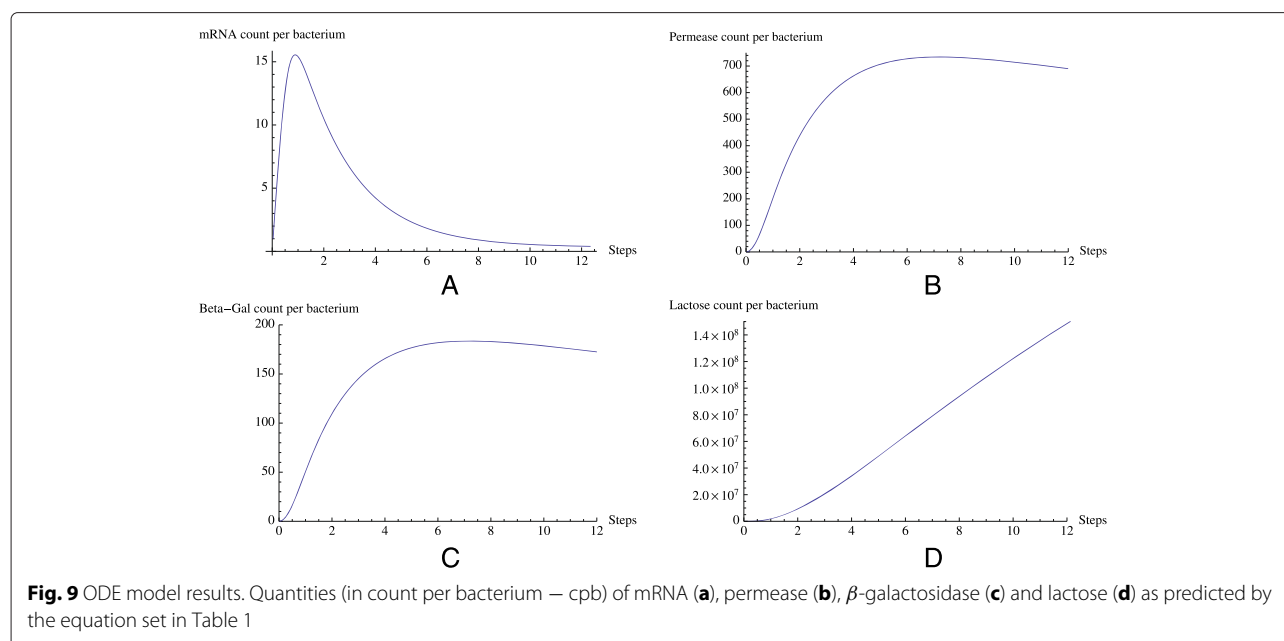
We consider the Santillan model complementary to our agent-based modeling approach. We will use the results from the Santillan model to validate our hybrid *Prokaryo* model, which we describe in more detail in the following section.

According to Santillan's model, the system is assumed to be glucose starved with an initial glucose concentration of 10 μ M. Lack of glucose promotes the formation of CAP complex which, in turn, aids RNA polymerase in transcribing the DNA. The initial external lactose concentration (L_e) is assumed to be 1000 μ M. The presence of lactose and allolactose will result in a conformational change in the repressor's structure causing it to detach from its position on the DNA (Fig. 6). The promoter is enhanced by the presence of CAP complex, becomes available, and is therefore no longer blocked by the repressor (Fig. 7). Thus, RNA polymerase begins transcription of DNA, and mRNA is generated. The resulting change in mRNA counts is illustrated in Fig. 9a. Ribosomes translate mRNA into peptide chains, which fold into permease and β -galactosidase. Figure 9b shows the average permease count per bacterium. The model assumes that an average of 750 permeases participate in transporting lactose into the cell. This causes the lactose count inside the cell to increase (Fig. 9d).

We have implemented the Santillan model into our *Prokaryo* simulation framework using the GNU Scientific Library (GSL), an open source numerical library in C and C++ [30]. The ODEs (Table 1) were solved using the GSL differential equation solver. In Fig. 9 we summarize the resulting concentration changes of mRNA, permease, β -galactosidase and internal lactose, as captured by the Santillan model. These plots do coincide with the results reported in the original paper [5].

Agent-based model of the lac operon

Although research in biological domains traditionally focuses on exploring interactions by means of equation-



based modeling, agent-based models are finding their way into more computational models of biological and biomedical systems. The research community has explored agent-based modeling in areas such as cancer research [31], immunology [32], clinical studies [33], vascular modeling [34] and developmental systems [35]. Agent-based modeling (ABM), also known as *individual-based modeling* (IBM), simulates interactions of agents with each other and their environment. Local interactions among agents give rise to complex global patterns, also known as emergence [36]. These effects are usually not visible by inspecting individual agents alone: the whole — through the network of interactions — is more than the sum of its parts.

Jacob and Burleigh in 2004 [37] built a spatial three-dimensional agent-based model of the lac operon. Their model treated each protein as an autonomous agent that interacts with other agents and their environment based on physical collisions. They modeled the lac operon on a double-helix plasmid with the Watson-Crick complementary pattern that closely mimics (part of) the genetic structure of the bacterial DNA. Jacob and Burleigh's model simulated the repressor protein and β -galactosidase (LacZ) effects on lac operon dynamics. However, the effect of permease (LacY), a transmembrane protein which helps to import lactose into the cell, was ignored. Jacob and Burleigh's model also abstracts the effect of catabolite repression on the system. Rather, they focus on key gene regulatory interactions of the lac operon. With *Prokaryo*, we took a similar approach to Jacob and Burleigh's model and built an extended 3-dimensional agent-based model of the lac operon.

LINDSAY composer: an ABM-ODE hybrid lac operon model

Although agent-based models can replicate emergent phenomena from agent interactions, simulating collective behaviours of individual agents is not without cost. A major limiting factor is the large number of interacting agents that need to be taken into account. The number of interactions that need to be tracked increases exponentially with the number of agents. In the lac operon example, a true multi-agent model would have to capture billions of interactions. Thus, it is computationally challenging to capture all the agent interactions, and run the model in real time to offer user interaction at all times. As interactivity is one of our main objectives, we needed to find a compromise between model accuracy and model responsiveness.

One way to address the high computational cost is through differential equations, which can inform the ABM of changes in the numbers of agents over time. However, differential equations are not the best choice to capture (1) system dynamics with spatial constraints, (2) when the number of reacting agents is low and (3) when the system is sensitive to small perturbations. All these aspects play a role in most biomolecular interactions, and are particularly prominent in the lac operon regulatory system.

We added an agent-based modeling framework to the differential equation set described above as the Santillan model [5]. This hybrid modeling approach combines continuous mathematical models, to capture large scale changes in biomolecular concentrations, with discrete agent interactions among smaller numbers of entities (a few to thousands). In order to implement a cellular model that incorporates both mathematical and agent-based simulations of a bacterium, we use a physics enabled

simulation platform called *LINDSAY Composer*, which we developed in our lab [38]. *Composer* combines high-end 3D graphics libraries [39] with a physics engine [14] and provides a (game) development environment for building visually appealing, semi-realistic agent-based models. Through hierarchies of interconnected ‘components’, *Composer* provides the necessary software infrastructure to build, run, and visualize customized simulations and interact with them in real time. *Composer* has been used to also build models of immune system processes [40], blood coagulation [41], and developmental systems [42]. Videos that illustrate *Composer* functionality and the graphical user interface for real-time interaction are available on the *LINDSAY* website (http://www.lindsayvirtualhuman.com/?page_id=469). The programming interface for *LINDSAY Composer* is similar to currently available game development software, such as Unity 3D (<http://unity3d.com>) or Unreal Engine (<http://unrealengine.com>).¹

Hybrid model architecture

The processes of transcription, translation, and protein folding are modelled through agents. Transcription and translation is performed using agents which represent RNA polymerases, mRNAs, and ribosomes. Peptide chains, which are also represented as agents, fold into proteins.

The quantities for lactose transport into the cytoplasm and lactose metabolism are modeled using the continuous ODE system (Table 1). The numbers generated from the mathematical model, in turn, inform the display components of *Prokaryo* from which the visual representations for the agents and particles that represent lactose, glucose, and galactose are generated. This functional division of our hybrid model is illustrated further in Fig. 10.

In order to closely mimic actual protein structures within the simulated cell, we use mesh representations of the 3-dimensional protein structures from the Protein Data Bank (PDB), the world’s largest repository of protein structural data [43]. More precisely, our *Prokaryo* model employs the following entities: the *E. coli* cell, RNA polymerase, lac repressor, CAP (cAMP-CRP) complex, mRNA, ribosome, permease, β -galactosidase, glucose, galactose, water molecules, allolactose and lactose. Figure 11 depicts the 3-dimensional representations for each of these agents.

Coding the biomolecular agents

In this section we use pseudo-code to describe the implemented behaviour for each of our key agents in the *Prokaryo* model: repressors, RNA polymerases, messenger RNAs, ribosomes, and polypeptide chains. For the remaining sections, we do not explicitly add the term

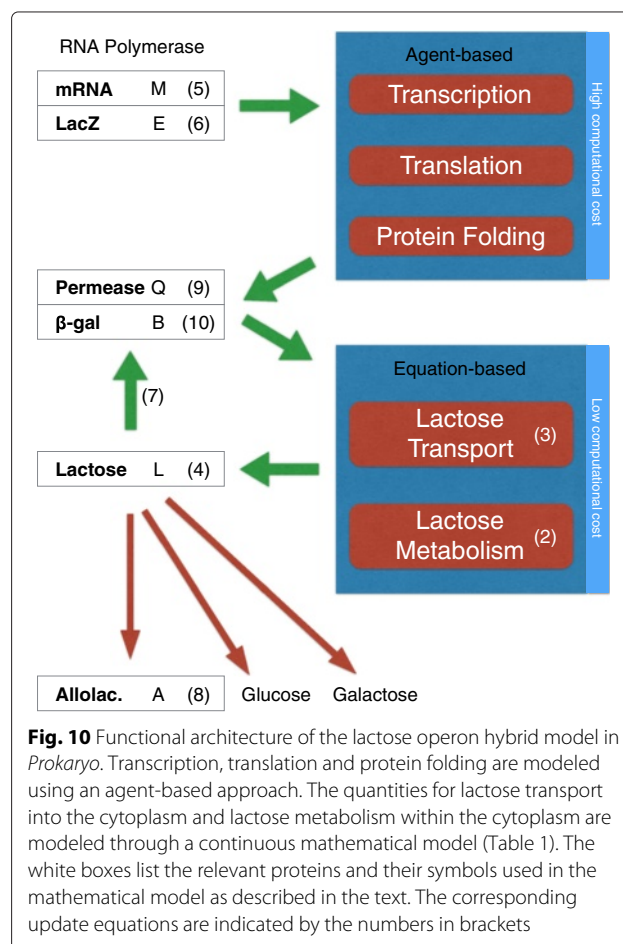


Fig. 10 Functional architecture of the lactose operon hybrid model in *Prokaryo*. Transcription, translation and protein folding are modeled using an agent-based approach. The quantities for lactose transport into the cytoplasm and lactose metabolism within the cytoplasm are modeled through a continuous mathematical model (Table 1). The white boxes list the relevant proteins and their symbols used in the mathematical model as described in the text. The corresponding update equations are indicated by the numbers in brackets

‘agent’ any more, if the context is clear. For example, we refer to the ‘repressor agent’ simply as ‘repressor’, the ‘RNA polymerase agent’ as ‘RNA polymerase’, etc.

Repressor: agent representation and coding

The repressor (Fig. 11c and d) is implemented as an agent that can be in the following three states according to Algorithm 1:

- **WANDERING:** The repressor is randomly moving within the cytoplasm.
- **DOCKED:** Upon collision with the operator-promoter region, the repressor undergoes a change in state from WANDERING to DOCKED, after which it blocks RNA polymerase from transcribing the operon.
- **INACTIVE:** The repressor has undergone conformational change by the presence of allolactose attached to its active site (Fig. 11d). The repressor will undock from its operator-promoter region and will not participate in the blocking of transcription initiation by RNA polymerase.

The conformational change of the repressor upon the docking of allolactose is depicted by a colour change

Algorithm 1 Repressor

```

1: CREATE:
2: state becomes WANDERING

3: ITERATE:
4: if state is WANDERING then
5:   if collided with promoter then
6:     dock to promoter
7:     set state to DOCKED
8:   else
9:     move randomly within the cell
10:  end if
11: else if state is DOCKED then
12:   if allolactose has docked on active site: then
13:     set state to INACTIVE
14:     undock from the DNA
15:   else
16:     do nothing
17:   end if
18: else if state is INACTIVE then
19:   if undock successful then    ▷ One second delay
20:     set state to WANDERING
21:   else
22:     wait until the next iteration
23:   end if
24: end if

```

(Fig. 6). Note that in Line 19 of Algorithm 1, after the repressor undocks from the promoter, it needs to move away from the promoter for at least one second so that the undocking event is recognized. This is due to the nature of the physics engine which registers rigid body collisions every 0.3 seconds, an internal parameter of the *Bullet Physics* simulation engine [14].² Without an enforced one second delay, due to overlapping rigid bodies the physics engine would register another collision within 0.3 seconds after the repressor is undocked.

RNA Polymerase: agent representation and coding

RNA polymerase (Fig. 11a) is the agent which initiates the transcription of the lac operon regulatory mechanism (Algorithm 2). RNA polymerase moves randomly within the cytoplasmic space of the cell. Upon approaching the DNA and colliding with the promoter section of the operon, RNA polymerase attaches itself to the promoter-operator section. In an attempt to initiate transcription, RNA polymerase starts scanning the DNA codons. If a repressor is actively blocking the promoter-operator site, RNA polymerase can no longer proceed and will undock from the DNA. In case the promoter-operator region is not blocked by a repressor, RNA polymerase initiates transcription. While RNA polymerase transcribes the genes on the DNA, it releases a newly formed mRNA strand

which detaches once the polymerase reaches the stop codon at the end of the operon (Fig. 7a-f).

As outlined in Algorithm 2, RNA polymerase is an agent implemented as a finite state machine with the following internal states:

- **WANDERING:** Polymerase randomly moves inside the cytoplasm.
- **TRANSCRIBING:** Polymerase has encountered a promoter section and attaches to DNA. Transcription is initiated and polymerase can be in one of the following sub-states:
 - **ON_PROMOTER:** Polymerase is docked on the DNA and starts scanning the promoter moving toward the operator.
 - **ON_OPERATOR:** Once polymerase reaches an operator after scanning the promoter, it will change its transcribing state to ON_OPERATOR and moves along the DNA. If a repressor is docked on the operator, polymerase either undocks from the DNA or initiates transcription based on T_a , the transcription initiation affinity constant, $0 \leq T_a \leq 1$ (Table 3). In *E. coli* multiple repressors constantly dock and undock from the promoter-operator region. Between these repressor docking and undocking events, RNA polymerase might attach and initiate transcription. Before docking, each RNA agent generates a random number $0 \leq \tau \leq 1$. Transcription occurs for $\tau \leq T_a$.
 - **ON_LACZ:** Polymerase starts transcribing the lacZ gene and generates an mRNA as it moves along toward the lacY gene.
 - **ON_LACY:** Polymerase has finished transcribing lacZ and is initiating transcription of lacY. mRNA elongation is continued as the polymerase scans along lacY.
 - **ON_LACA:** Polymerase successfully transcribed lacY and is initiating transcription of lacA.
- **UNDOCKING:** Either polymerase has reached the stop codon at the end of the lacA gene or has encountered a repressor at the promoter-operator region, thus detaching itself from the DNA.

One of the major challenges of simulating cell dynamics is the time mapping between *in vivo* to *in silico*. To realistically model these dynamics, a scale factor is necessary to speed up slow processes or slow down fast processes. It takes 8.23 seconds (on average) for the RNA polymerase agent *in silico* to fully transcribe the lac operon. Based on the length of the lac operon (4941 base pairs [44]) and

Algorithm 2 RNA Polymerase

```

1: CREATE:
2: state is set to WANDERING
3: subState is set to NULL      ▷ This state used only when
   transcribing.
4: set SELF_TRANSCRIPTION_AFFINITY to random number
   between 0 and 1.

5: ITERATE:
6: if state is WANDERING then
7:   if encountered promoter then
8:     dock to the promoter
9:     set state to TRANSCRIBING
10:    set subState to ON_PROMOTER
11:   else
12:     randomly move in cytoplasm
13:   end if
14: else if state is TRANSCRIBING then
15:   if subState is ON_PROMOTER then
16:     scan the next codon
17:     if reached end of promoter sequence then
18:       set subState to ON_OPERATOR
19:     end if
20:   else if subState is ON_OPERATOR then
21:     if next codon is blocked by repressor then
22:       if SELF_TRANSCRIPTION_AFFINITY ≤
TRANSCRIPTION_AFFINITY then
23:         move passed the repressor
24:       else
25:         set state to UNDOCKING
26:       end if
27:     else
28:       scan the next codon
29:       if reached the end of operator sequence then
30:         create an empty mRNA molecule
31:         set subState to ON_LACZ
32:       end if
33:     end if
34:   else if subState is ON_LACZ then
35:     scan the next codon
36:     append codon to mRNA
37:     if reached end of the LacZ then
38:       set subState to ON_LACY
39:     end if
40:   else if subState is ON_LACY then
41:     scan the next codon
42:     append codon to mRNA
43:     if reached end of the LacY then
44:       set subState to ON_LACA
45:     end if
46:   else if subState is ON_LACA then
47:     scan the next codon
48:     append codon to mRNA
49:     if reached end of the LacY then
50:       set state to UNDOCKING
51:     end if
52:   end if
53: else if state is UNDOCKING then
54:   set subState to NULL
55:   release constructed mRNA
56:   undock from the operon
57:   set state to WANDERING
58: end if

```

the maximal *in vivo* RNA transcriptional speed (approximately 60 base pairs per second [45]), on average it would require 82.35 (4941/60) seconds for transcription to complete. Every second in the simulation is mapped to 10

seconds *in vivo* (Table 3). Thus, the time required for the RNA agent to complete transcription is 8.23 seconds which corresponds to 82.3 seconds within a real *E. coli* bacterium [1].

Messenger RNA: agent representation and coding

mRNA (Fig. 12) is an agent that does not exist at the beginning of the simulation. This is because mRNA is dynamically generated in real time upon transcription of the operon. mRNA generation is initiated by transcription of lacZ. As RNA polymerase transcribes the three genes of lac operon, mRNA elongation occurs (Figs. 13 and 14).

mRNA is treated as an agent with the following properties and states:

- **Lifetime:** upon generation of an empty mRNA by RNA polymerase, each mRNA is assigned a random lifetime between 1 to 6 minutes which is equivalent to 6 to 36 seconds in the simulation. The range is chosen to mimic mRNA lifetime in bacterial cells [46, 47].
- **Age:** is a parameter determining how long mRNA has existed within the system. Once mRNA is generated, its age is initialized to zero, $age = 0$. At every simulation step, age is incremented by a value of ΔT : $age = age + \Delta T$.³ This parameter is used as part of the mRNA degradation where mRNA is removed from the cell once $age == lifetime$.
- **Availability State:** During mRNA generation (elongation), multiple ribosomes dock on the mRNA to initiate translation. Ribosomes can translate the mRNA simultaneously. In order to avoid collisions between translating ribosomes, there has to be a gap between translation initiation (ribosome spacing). The availability parameter enforces this gap between ribosomes docking on the mRNA and allows for a minimal number of nucleotide distance between each translating ribosome. After a ribosome has docked, the availability state of mRNA becomes UNAVAILABLE. This will block other ribosomes from docking onto the mRNA to avoid collisions between ribosomes translating the mRNA [47]. Once the current ribosome has progressed, mRNA becomes AVAILABLE for the next ribosome to initiate transcription.

In *E. coli*, ribosomes are readily available. In this study, to simplify interactions and not visually overwhelm the user with ribosomes that block visualizing the whole process, ribosome agents are created by mRNA agents. This is based on the assumption that at any point in time there are numerous ribosomes colliding with mRNA.

Ribosome: agent representation and coding

The cell is filled with hundreds of ribosomes (Fig. 11b) which are constantly seeking an mRNA to translate. Once

Algorithm 3 Messenger RNA

```

1: CREATE:
2: age is set to 0
3: lifeTime is set to a random number between 1 to 6
   (minutes)
4: state is set to AVAILABLE

5: ITERATE:
6: if age  $\geq$  lifeTime then
7:     destroy self
8: else if state is AVAILABLE then
9:     create a ribosome
10:    attach ribosome to first codon
11:    set the state to UNAVAILABLE
12:    age = age + deltaT
13: else if state is UNAVAILABLE then
14:    if ribosome gap is sufficient then
15:        set state to AVAILABLE
16:    else
17:        wait till next iteration
18:    end if
19:    age = age + deltaT
20: end if

```

a ribosome collides with a proper section of an mRNA (based on the availability state of the mRNA) it will initiate translation. As translation progresses, ribosomes use the genetic information to generate an appropriate peptide chain. Upon completion of translation, this peptide chain will fold into a protein.

Once a ribosome agent is created by an mRNA agent, it gets attached to the first codon. The ribosome then becomes active and initiates translation. It will traverse the mRNA's codon one by one and add the appropriate amino acid sequences (based on the mRNA codon) to the peptide chain. The ribosome generates a separate peptide chain for each gene which later folds into a protein. Peptide chain elongation occurs during translation of a specific gene. The chain is detached once the ribosome moves on to the next codon corresponding to the next gene in the sequence. Once the ribosome reaches the end of an mRNA, the last polypeptide chain it has generated will detach and start folding into a protein.⁴ The ribosome agent is removed from the simulation after it has finished its mRNA transcription.

In a real cell, after translation of an mRNA strand, ribosomes become readily available for translation again. In our model, once a ribosome completes translation, it is removed from the simulation. This simplification does not affect the system due to the fact that ribosomes continuously dock and translate mRNA.

Much like the challenge presented for RNA polymerase transcription, translation by the ribosome in *E.coli*

Algorithm 4 Ribosome

```

1: CREATE:
2: start a new polypeptide chain

3: ITERATE:
4: if reached the end of mRNA then
5:     detach polypeptide chain
6:     destroy self
7: else if reached the end of current gene codon on
   mRNA then
8:     detach polypeptide chain
9:     start a new polypeptide chain
10:    move to the next codon
11: else
12:    add an amino-acid to the polypeptide chain
13:    move to the next codon
14: end if

```

proceeds at a maximum speed of about 20 aa/sec (amino acids per second) [48]. Given that there are 4941 base pairs and every 3 base pairs code for one amino acid, the number of amino acids is 1647 (4941/3). Translating the amino acids at 20 aa/sec results in a total of 82.35 seconds to complete translation (1647/20). Again, considering the simulation time in relation to real time, 82.35 real time seconds map to 8.23 (82.35/10) seconds in the simulation.

Algorithm 5 Polypeptide Chain

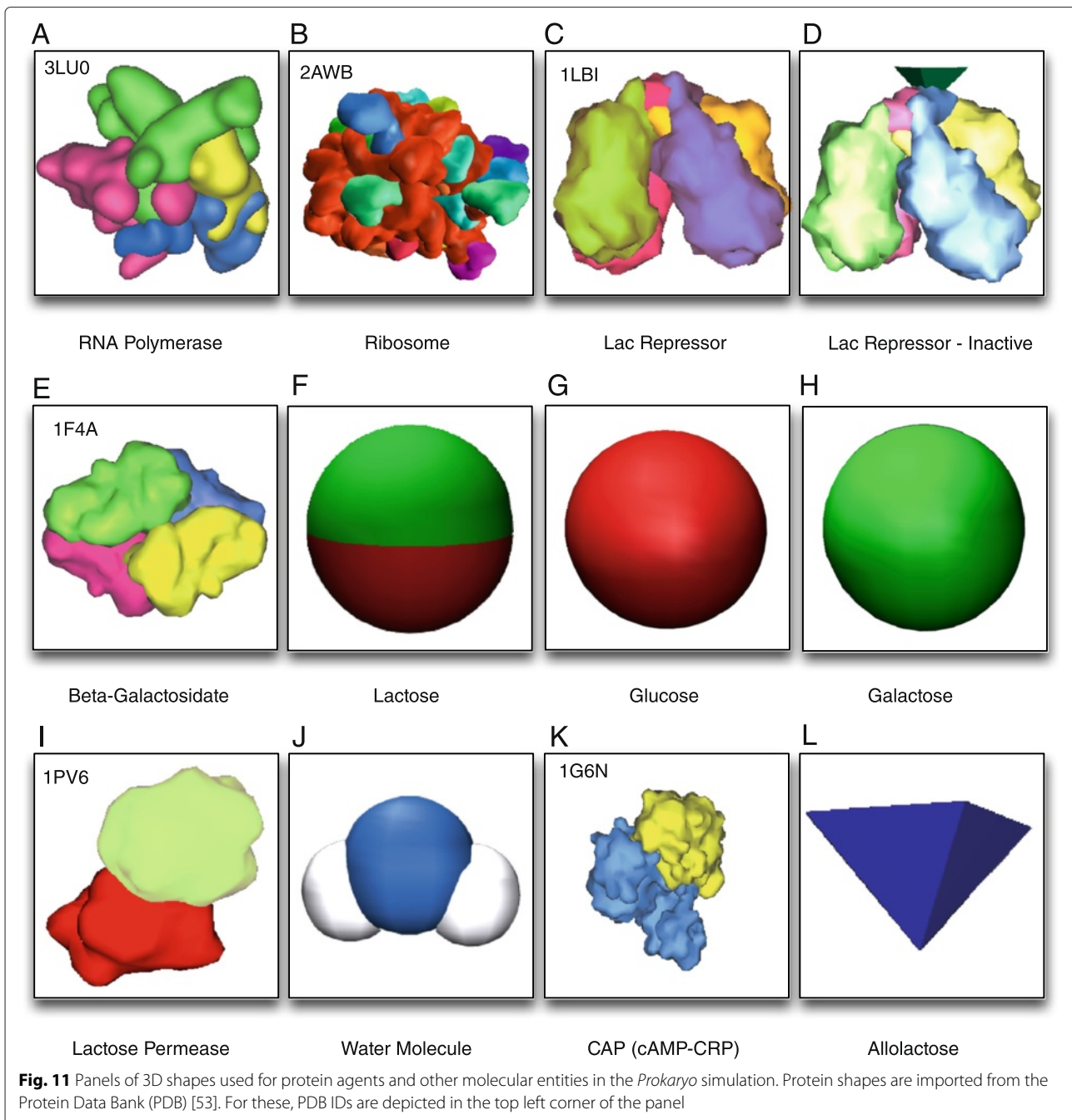
```

1: CREATE:
2: set state to NOT-FOLDED

3: ITERATE:
4: if detached from ribosome then
5:     if state is NOT-FOLDED then
6:         fold the chain (FOLD:)
7:     else if state is FOLDED then
8:         if should replace chain with protein then
9:             remove folded chain
10:            insert protein mesh
11:        end if
12:    end if
13: end if

14: FOLD:
15: randomly select pairs of amino acids in the chain
   based on the chain's length
16: introduce a connection (peptide bond) between
   amino acid pairs
17: enable the connections  $\triangleright$  : this causes the chain to
   fold based on physics constrains among pairs
18: set state to FOLDED

```



Interestingly enough, this is the same time span required for RNA polymerase to transcribe DNA.

Polypeptide chain: agent representation and coding

As the name suggests, the polypeptide chain is composed of a chain of peptides joined together that provide a backbone for protein molecules. The peptide chains are obtained from amino acids that fold into proteins. The details of protein folding are beyond the scope of this paper. Thus, to simplify the folding process and yet mimic

the structural formation of amino acids in real *E. coli*, a simplified folding mechanism of chains is introduced. Figure 14a and b depict a ribosome translating a mRNA generating a peptide chain. Later this chain will fold into a protein (Fig. 14c).

Once generation of a polypeptide chain is completed, the chain detaches from the ribosome. Once the chain is detached, a random number of peptide bonds (based on the length of the chain) is introduced among randomly chosen pairs of amino acids (Fig. 8).

Table 3 List of key parameters used in the *Prokaryo* model: parameter values have been set in order to replicate experimental data as reported from wet lab experiments and in the literature [1]

Parameter	Description	Value
n	Nucleotides per unit	3
Tr	Transcription speed	60 base pairs per second
Gt	Glucose threshold	40000 count per bacterium
Ro	Ribosome offset	0.1
Tf	Translation frequency	0.25 sec^{-1}
Rd	Ribosome dock delay	1.0 sec^{-1}
Ptc	Protein threshold count	3.0
Rlt	Repressor lactose threshold	2000 cpb
Ta	Transcription affinity	0.01
St	simulation time factor	10

The *Prokaryo* E. coli cell as a hybrid model

Lac operon regulation occurs within the bacterial cytoplasm. The agents and entities involved in gene regulation (Fig. 11) interact and communicate with each other in specific regions within the cell. Inspired by Goodsell’s illustrations [1], our intention is to visualize the ‘crowdedness’ of a cell. For the actual cell body we used a custom-built, 3-dimensional mesh model of an *E. coli* bacterium cell body [49]. The *E. coli* model consists of a double layer membrane, a flagellum and pili on the surface of the outer membrane. The long, tubular structure within the cytoplasm represents the DNA as has been generated by us procedurally. The cell does not act as an agent, but rather represents a container for all agents within the cytoplasm.

The cell environment itself is divided into three subsections: the space outside (relative to the bacterium), the periplasm (between membranes) and the cytoplasm (inside the cell). Figure 15 shows screenshots of a realtime fly-through from outside the cell into the cytoplasmic space.

Occupying most of the cytoplasm, *E. coli*’s DNA is represented as a thin, tubular structure. The procedurally generated DNA structure covers an appropriate portion of the cytoplasm and is to scale relative to its *E. coli* body and its internal proteins. Instead of a double-helix, DNA

is represented as a more simplistic shape, which facilitates the simulated physics calculations. On the DNA, a specific segment is chosen to represent those sections that are involved in the lac operon (Figs. 3 and 5). The section representing the operon is subdivided into colour coded segments for promoter, operator, LacZ, LacY and LacA genes, respectively.

ABM-ODE model switching

The agent-based model captures transcription, translation and protein folding (Fig. 10).⁵ Once a peptide chain folds, the folded structure is replaced with a three-dimensional model of the respective protein: permease or β -galactosidase (Fig. 11). The agent model communicates and synchronizes with the mathematical equations to compute the number of agents involved in lactose metabolism based on β -galactosidase production and lactose transport rate, which is based on permease production. Based on the number of β -gal protein agents, the equation based model computes the rate at which lactose is metabolized. Similarly, permease counts are used to compute the rate at which lactose is transported into the cell.

More precisely, we increment the β -galactosidase count every time a corresponding peptide chain is folded. The rate l_{mtbol} of lactose being broken down per β -galactosidase protein is [5]:

$$l_{mtbol} = \frac{2(\phi_M \times L)}{k_M + L} \tag{1}$$

Here, L represents internal lactose; k_M is the maximal transcription initiation rate of the promoter (2.0 sec^{-1}), and ϕ_M is the maximum rate of lactose-to-glucose metabolism ($3.6 \times 10^4 \text{ min}^{-1}$) (Table 4). Consequently, the number of lactose entities, L_{mtbol} , metabolized per ΔT with b entities of β -galactosidase present is:

$$L_{mtbol} = l_{mtbol} \times b. \tag{2}$$

The rate of lactose influx per permease (p) is set to $p_{lac} = 1000$ lactoses per ΔT per permease, from which we calculate the number of lactoses transported into the cytoplasm per ΔT as [5]:

$$L_{trans} = p_{lac} \times p. \tag{3}$$

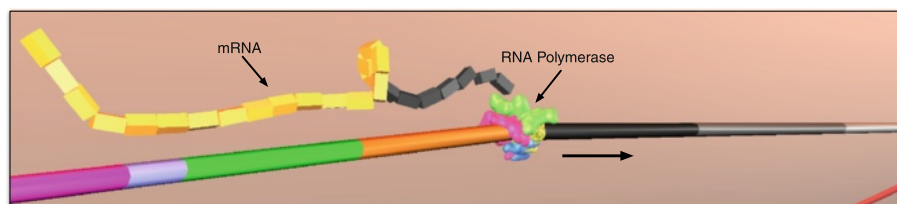


Fig. 12 An RNA Polymerase transcribing the DNA generating an mRNA. Each colour coded section on the DNA refers to a specific gene. Direction in which transcription occurs is depicted by the arrow

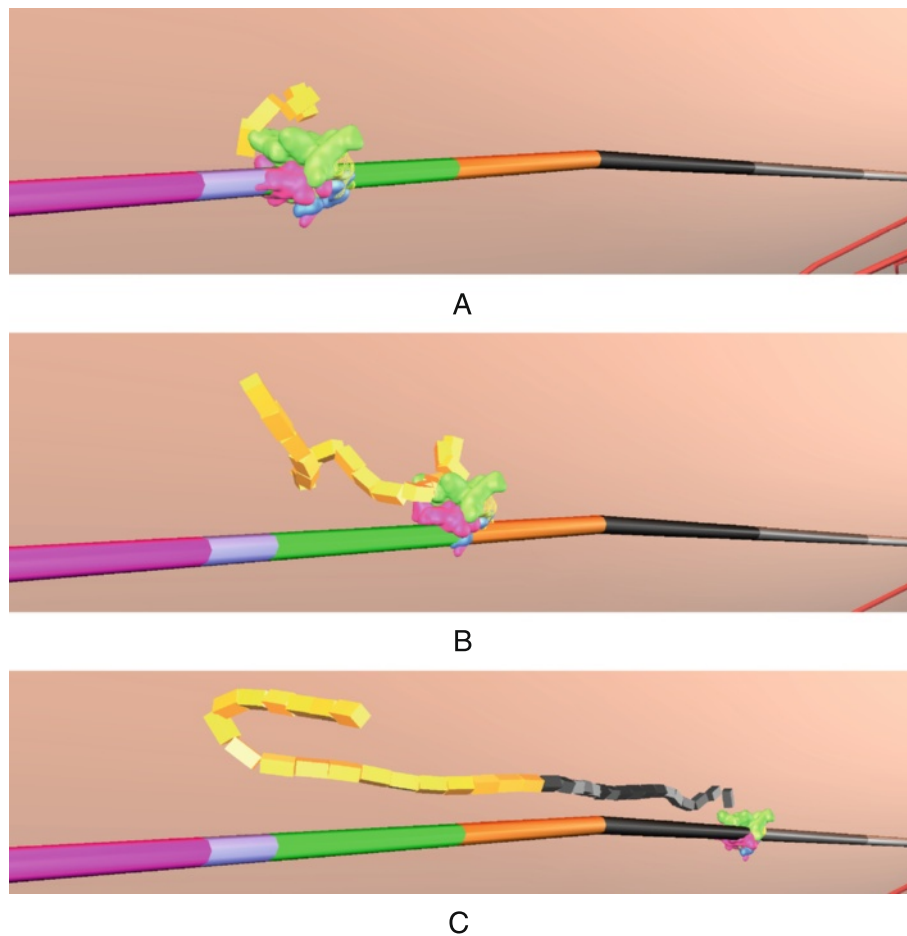


Fig. 13 DNA transcription by RNA polymerase resulting in the generation (elongation) of messenger RNA. Transcription initiation is depicted in **a**. As the transcription progresses (**b**), mRNA elongation occurs (**c**)

From the above equations, we can update the total number of lactoses L in the cytoplasm:

$$L := L - L_{mtbol} + L_{trans}. \quad (4)$$

The lactose counts feed into the particle systems (described below) to visualize these entities in realtime, while keeping the simulation interactive (Fig. 1). A visual representation of lactose metabolism by β -galactosidase is depicted in Fig. 4. In order to simulate lactose transport through transmembrane permeases, an agent-based representation of this process has been implemented as well (Fig. 3).

Visualizing a crowded cell space

A bacterial cell is filled with billions of water molecules, proteins, and other macro and micro molecules. Due to the critical role of proteins, each protein agent in the simulation is modeled as an individual physical entity. This means the agent has physical properties and interacts

with its environment through physical collisions that trigger reactions and state changes in the system. Simulating physical interactions is computationally very expensive, though. Even with today's powerful computers, their high computational capabilities (e.g., by utilizing GPU algorithms) and despite sophisticated physics simulation libraries (e.g., Bullet [14]), only a relatively small subset of participating entities can be considered for a physical simulation. However, only a very small subset of interacting molecules and proteins actually have direct effects on the lactose operon. In other models, we have used an 'abstraction' mechanism to alleviate the computational challenges with large, multi-agent simulations [41, 50]. The basic idea is to observe agent dynamics, extract interaction patterns, and temporarily replace groups of agents by a single 'cluster agent'. The validity of this 'abstraction' needs to be checked on a regular basis and, if necessary, all agents of a cluster will be released back into the simulation. In *Prokaryo*, however, we use yet another technique to visualize large numbers of entities

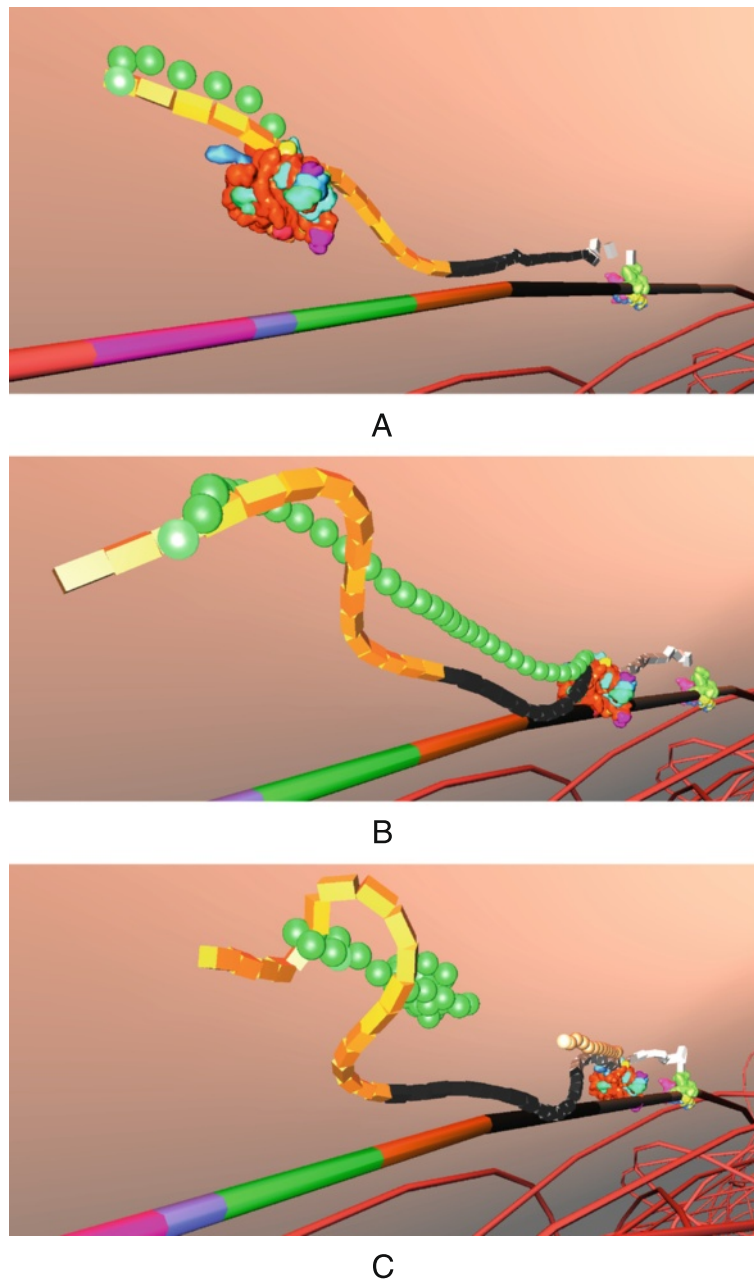


Fig. 14 Ribosomal translation of mRNA. The ribosome scans and begins translation of the mRNA (a). Peptide chains are formed (b) and the chain begins to fold (c)

(Fig. 1): particle systems, which we explain in the next section.

Particle systems

In addition to proteins, other entities that protein agents interact with need to appear in large quantities (e.g., lactose). To simulate the crowded universe of biomolecules within a cell, we use a computer graphics technique known as *particle systems* [51]. Particle systems

harness the power of graphics processors (GPUs) and are able to render small entities (particles) in large numbers. Particle systems have been used to simulate phenomena that consist of very small interacting elements such as water, sparks, clouds, fog or snow. With particle systems, we are able to visualize over 70,000 entities inside the cell, together with about 1,000 protein agents with actual physical properties, while maintaining full interactive control of the simulation.

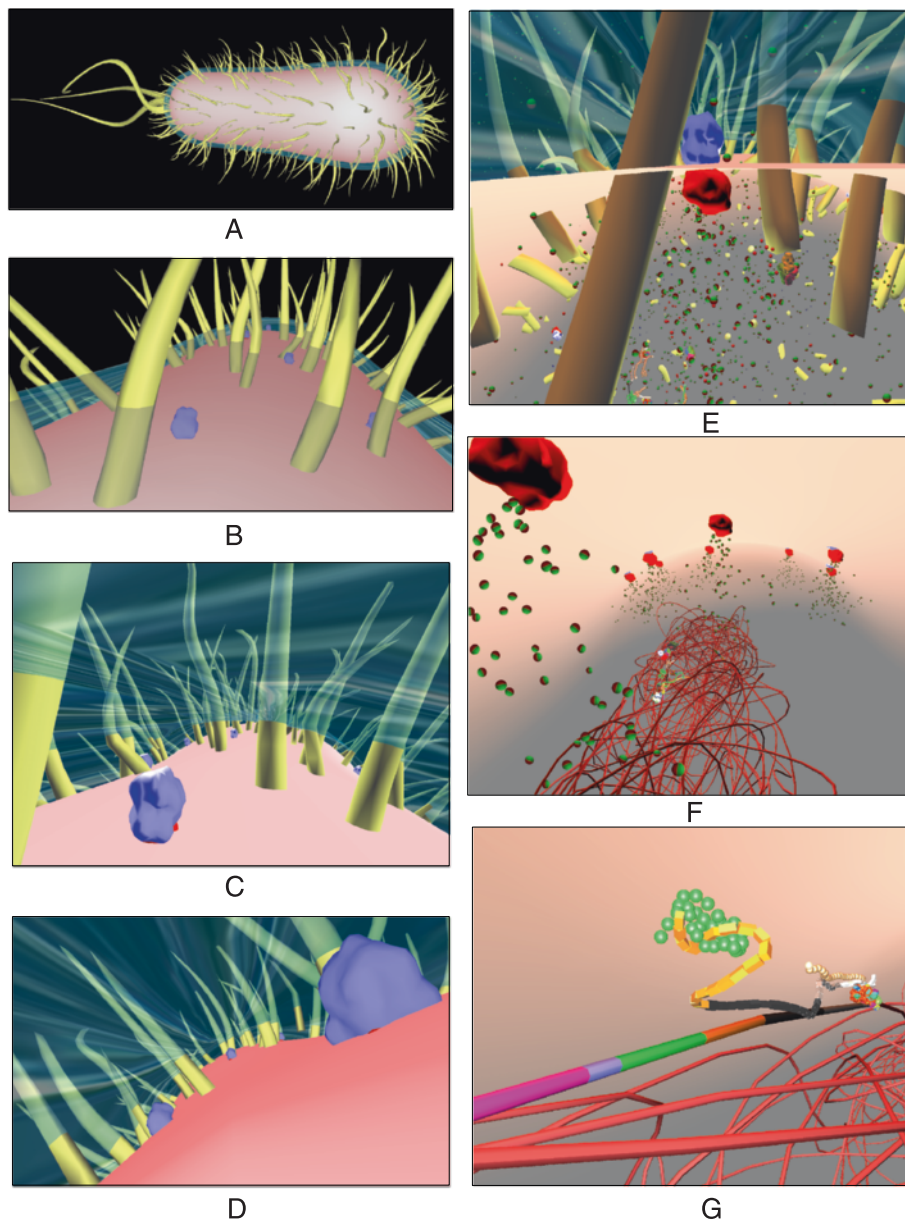


Fig. 15 A *Prokaryo* Fly-through. The snapshots are taken while navigating through the virtual cell model. **a** approaching the cell from the outside; **b** close to the membrane, with pili and lactose permease pumps visible; **c** and **d** just below the periplasm between membranes; **e** focus on lactose permease, with lactose and other molecules in the background; **f** cytoplasmic space with coiled DNA structure in the center. **g** close-up of the lactose operon sections on the DNA with mRNA and folding amino acid chains. For this fly-through, most proteins and other entities inside the cell are not depicted. The rendering and hiding of specific molecules is controlled by the user. Hidden molecules are not visible, yet remain active elements in the simulation, as depicted in Fig. 1

Results and discussion

Now that we have discussed the mathematical, computational, and rendering aspects of our modeling framework, we present a step-by-step description of our *Prokaryo* simulation. We explain which aspects are captured by our model and how the generated outcomes match the actual biomolecular processes inside an *E. coli* cell.

Repressor

At the beginning of the simulation, the cell contains no glucose. Lactose is located outside the cell, with some lactose undergoing passive transport into the cell. As lactose concentration is below the *Rlt* threshold (Table 3), repressor is docked on the promoter, thus blocking transcription of DNA (Fig. 6). Each repressor undocks

Table 4 List of constants used in our reproduced model (compare Table 1)

Lactose operon model parameter list		
Parameter	Value	Description
μ	0.02 min^{-1}	Bacterial growth rate.
D	2 mbp	Lac promoter concentration.
k_M	180 min^{-1}	Maximal transcription initiation rate of the lac promoter.
k_E	18.8 min^{-1}	Translation initiation rate of lacZ transcript.
k_L	$6.0 \times 10^4 \text{ min}^{-1}$	Maximal lactose uptake rate per permease.
γ_M	0.48 min^{-1}	lacZ mRNA dilution/degradation rate.
γ_E	0.03 min^{-1}	Lac permease degradation/dilution rate.
γ_L	0.02 min^{-1}	Lactose degradation/dilution rate.
k_{pc}	30	Cooperative promoter-CAP binding site interaction
p_p	0.127	Polymerase binding probability to the lac promoter.
ϕ_M	$3.6 \times 10^4 \text{ min}^{-1}$	Max. rate of lac-to-allolac and lac-to-gal metabolism
K_G	$2.6 \mu M$	CAP complex binding affinity to DNA based on external glucose concentration.
n_h	1.3	CAP complex binding affinity to external glucose.
ξ_2	0.05	Affinity of active repressor for Operator O_2 .
ξ_3	0.01	Affinity of active repressor for Operator O_3 .
ξ_{123}	163	Stability of the O_1 - O_2 - O_3 -repressor complex.
ρ_{max}	1.3	Repression affinity of lac operon promoter
K_A	$2.92 \times 10^6 \text{ mpb}$	Allolactose-repressor subunit dissociation rate.
k_L	$680 \mu M$	Half-saturation constant for lactose uptake rate.
ϕ_G	0.35	Permease activity as a function of inside glucose concentration
k_G	$1.0 \mu M$	Permease activity as a function of outside glucose concentration.
k_M	$7.0 \times 10^5 \text{ mpb}$	Max. transcription initiation rate of lac promoter

with probability λ_{rep} . Consequently, multiple repressors are docking and undocking continuously. Between the undocking of one repressor and docking of another repressor, RNA polymerases may have a window of opportunity to attach to the promoter and initiate transcription with probability $\lambda_{poly} = 1/100$. For every 100 transcription initiation attempts, only one permease will be

successful. This value was determined through a number of simulation experiments. A lower transcription initiation rate would cause our system to halt. A higher number would result in RNA polymerases to initiate transcription and ignore the presence of repressor. Figure 16a illustrates the number of mRNAs resulting from transcription through polymerases. The basal level expression is labeled as Area B.

Transcription and translation

Unsuccessful attempts of transcription happen during the period marked as Area A. Once the operon is expressed at basal levels, resulting in mRNA copies, translation of permease and β -galactosidase is performed by ribosomes (Fig. 7). Permease acts as a transmembrane protein and actively transports lactose at a much higher rate than through passive transport alone. β -galactosidase metabolizes lactose into glucose and allolactose. Allolactose is subsequently metabolized into glucose and galactose. Allolactose deactivates the repressor, which detaches from the operator. The operon genes are now expressed at their highest level (Area C). The number of mRNAs triples in less than 4 minutes (real time). Once the system has reached the glucose threshold of 40,000 cpb, cAMP concentrations decrease with CAP no longer bound to DNA. This leads to a reduction in the rate of transcription (Area D).

Lactose permease

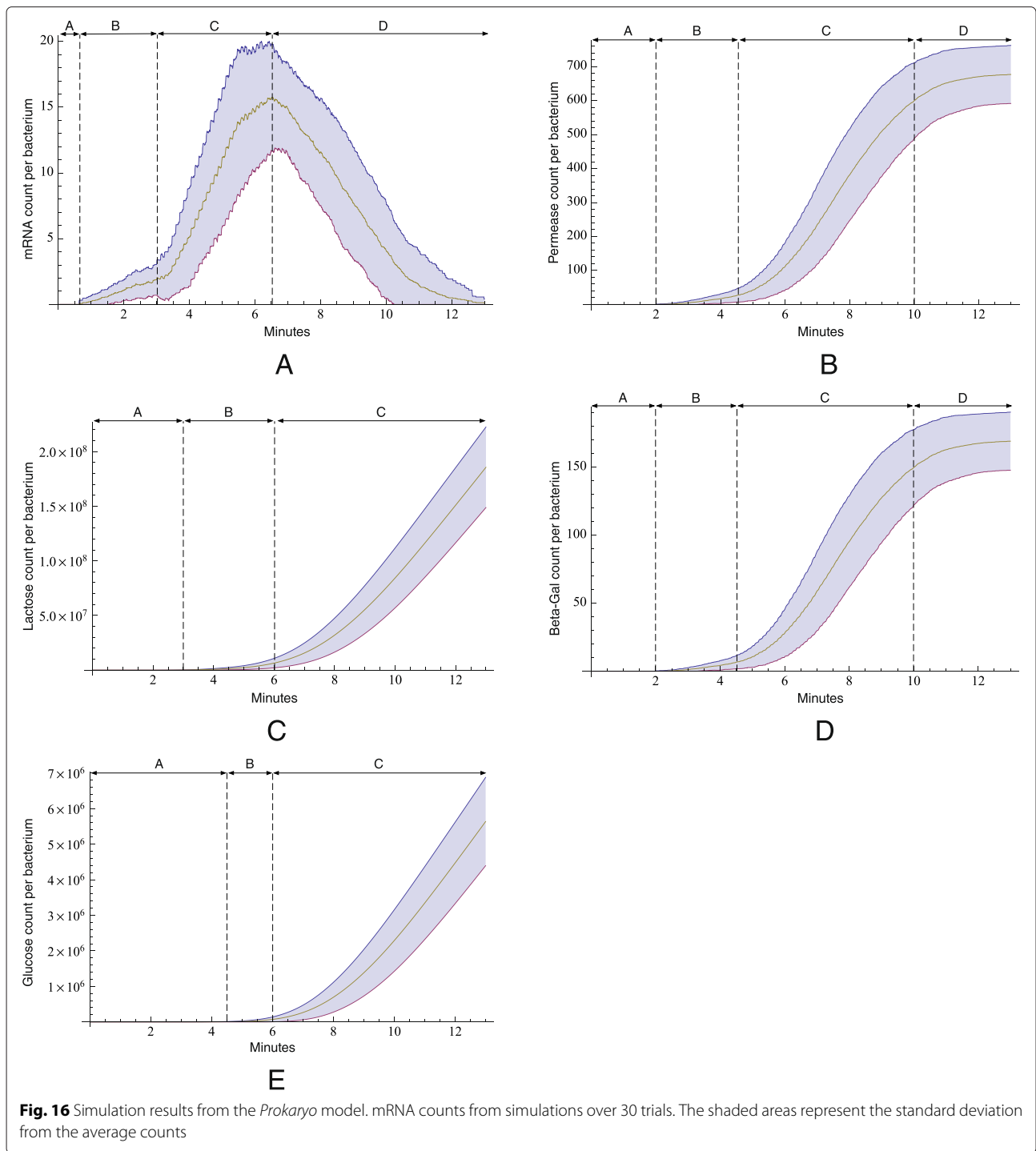
Figure 16b illustrates lactose permease counts over time. Area A represents the initial 2 minutes of unsuccessful transcription attempts by RNA polymerase. During base level expression (Area B), permeases are being expressed, starting to initiate lactose transport into the cell. In Area C, the operon is expressed fully, producing over 600 permeases within about 5 minutes. An equilibrium⁶ is reached once lactose production has stopped due to the absence of CAP complex on the DNA (Area D).

Lactose

It takes about 3 minutes until lactose starts to get imported into the cell (Fig. 16c, Area A). At base level expression, few active transmembrane proteins transport lactose into the cell, which initiates gene expression (Area B) via allolactose production. Once repressor is deactivated by allolactose, the lac operon genes are expressed at high levels. A rise in permease production leads to an exponential increase of lactose levels inside the cell (Area C). Even after gene expression decreases, the lactose count within the cell does not decrease. This is due to active permeases still transporting lactose into the cell.

β -galactosidase

It takes about 2 minutes before β -galactosidase is generated (Fig. 16d). Base level expression leads to small



production of β -galactosidase (Area B). Expression of lacY results in an increase in β -galactosidase (Area C). After the system reaches the glucose threshold, gene expression is minimized, thus halting production of β -galactosidase (Area D). The glucose threshold is set at 40,000 per bacterium. If the glucose threshold is considerably below this value, the lac operon switch is turned off too early. A

higher threshold, causes lactose metabolism and transport to increase drastically.

The lac operon facilitates the process to metabolize lactose when glucose levels are low. Glucose is the desired product, which requires a complex, multiple step process to produce. The time required by the lac operon to switch and initiate gene expression for glucose production



Fig. 17 Application on mobile devices. *Prokaryo* running on an iPhone, making use of the touch interface for easy navigation

is depicted in Fig. 16e (Area A). Base level expression results in a slight increase in the production of glucose (Area B). Once gene expression is activated by absence of repressor from the DNA, multiple polymerases transcribe the operon genes, which results in an increased count of mRNA. Consequently, more permease and β -galactosidase are produced (Fig. 16a and b). Permease transports more lactose into the cell. More β -galactosidase is expressed from the operon. This results in an exponential increase in lactose metabolism and, in turn, in increased glucose production (Area C).

Summary

The hybrid approach produces results that are in good alignment with the results from the continuous Santillan model [27]. Comparing mRNA predicted by the differential equations and the hybrid model, the trend of mRNA generation and degradation is very similar (Figs. 9a and 16a). On average, 700 permeases are generated to transport lactose into the cell as shown by both the differential equations and the hybrid model (Figs. 9b and 16b).

Both models predict the same rate of lactose metabolism (in orders of 10^8) as illustrated in Figs. 9d and 16c.

Conclusion

We have presented a hybrid model of *E. coli* metabolism related to the lactose operon. The dynamics of cellular 'agents' are defined by their rules of interaction, implemented as finite state machines. The agents are embedded within a 3-dimensional virtual environment with simulated physical and electrochemical properties. The hybrid model is driven by a combination of (1) mathematical equations (DEQs) to capture higher-scale phenomena and (2) agent-based rules to implement localized interactions among a small number of molecular elements. By reproducing results from *in silico* experiments reported in the literature, we have demonstrated that our hybrid model is able to capture phenomena across

multiple spatial scales, from changing concentration gradients to one-on-one molecular interactions.

Our article also demonstrates how interactivity and illustrations, in the form of detailed graphical representations in 3-dimensional scenes, can enhance the understanding of biological systems through exploration and visualization. We have implemented our *Prokaryo* model on desktop computers, but also on mobile devices with touch interfaces (Fig. 17). Mobile computing devices make such models more accessible. Touch interfaces greatly facilitate navigation through and exploration of such models.

Prokaryo is only our first 3D, interactive model of a larger biomolecular system. We will extend *Prokaryo* by including other pathways (such as chemotaxis) and embed *Prokaryo* into a colony of interacting cells. *Prokaryo* is part of the *Lindsay Virtual Human (LINDSAY)*, a multi-scale computational physiology simulation environment [38], in which we can place *E. coli* cells inside the *LINDSAY* intestinal tract. Any cell can be selected, and one can navigate into its 'cellular universe', which is what we have described here as *Prokaryo*. More information on the Lindsay Virtual Human can be found at <http://lindsayvirtualhuman.org>, including a video demonstration and executable download (http://www.lindsayvirtualhuman.com/?page_id=469).

Endnotes

¹Neither of these game development engines had been available at the time we started this project. In our most recent simulation projects we use both game engines to build our simulations.

²Physics engines are used extensively in computer games to simulate physical properties of 2- and 3-dimensional elements. Checking whether elements have collided and how such a physical interaction affects the colliding elements is implemented through physics engines.

³In our model, $\Delta T = 1/60$ second is the length of a simulation step as defined by the physics engine [14].

⁴We are aware that folding should occur earlier, but we have ignored this aspect to simplify our current model.

⁵Similar schemes for hybrid models have been developed recently. Most noteworthy is a hybrid model of bacterial ecology and chemotaxis [52]. We roughly follow the hybrid modeling framework proposed in this work. Compare, for example, Fig. 1 in [52] and Fig. 10.

⁶The permease count remains constant due to a permease degradation rate of zero.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

AE implemented the *Prokaryo* model, performed the described experiments, and wrote major parts of this article. TD developed *Composer* and provided guidance to AE with respect to the realization of the *Prokaryo* model. AW implemented the *Prokaryo* model system in our most recent *Composer* Version 2, which is available online. JA oversaw the biological aspects of this project. CJ presented the original idea, supervised the project, and composed the final version of this article. All authors read and approved the final manuscript.

Acknowledgements

This project was partly supported by an NSERC Discovery Grant from the Natural Sciences and Engineering Research Council, Canada.

Author details

¹Department of Computer Science, Faculty of Science, University of Calgary, 2500 University Drive NW, T2N 1N4 Calgary, Canada. ²Department of Biochemistry and Molecular Biology, Cumming School of Medicine, University of Calgary, 3330 Hospital Drive NW, T2N 4N1 Calgary, Canada. ³Department of Microbiology, Immunology and Infectious Diseases, Cumming School of Medicine, University of Calgary, 3330 Hospital Drive NW, T2N 4N1 Calgary, Canada.

Received: 12 March 2015 Accepted: 24 August 2015

Published online: 29 September 2015

References

1. Goodsell DS. The Machinery of Life, 2nd edn. Reading. New York, NY, USA: Springer; 2009.
2. Lue R. Biovisions. 2007. <http://biovisions.mcb.harvard.edu/>. Accessed Oct 2013.
3. Animation XS. Xvivo takes harvard university on a 3d journey. Technical report, Harvard University. 2006.
4. Abdelraheem A, Asan A. The effectiveness of inquiry-based technology enhanced collaborative learning environment. *Int J Technol Teach Learn*. 2006;2(2):65–87.
5. Santillan M, Mackey MC, Zeron ES. Origin of bistability in the lac Operon. *Biophys J*. 2007;92(11):3830–842.
6. Zimmer C. *Microcosm: E. Coli and the New Science of Life*. The Journal of Clinical Investigation. New York, NY, USA: Knopf Doubleday Publishing Group; 2008.
7. Cooper GM. *The Cell: A Molecular Approach*, 2nd edn. The Cell: A Molecular Approach. Sunderland, MA, USA: Sinauer Associates; 2000.
8. Alon U. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. London, UK: Chapman and Hall/CRC; 2007.
9. Jacob F, Monod J. Genetic regulatory mechanisms in the synthesis of proteins. *J Mol Biol*. 1961;3(3):318–56.
10. Beckwith J. The lactose operon. in *Escherichia coli and salmonella* In: Neidhardt F, Ingraham J, Low K, Magasanik B, Umberger E, editors. *Cellular and Molecular Biology*. Washington, DC, USA: American Society for Microbiology; 1987. p. 1444–1452.
11. Müller-Hill B. *The Lac Operon: A Short History of a Genetic Paradigm*. Berlin, Germany: Walter de Gruyter; 1996.
12. Oehler S, Eismann ER, Kramer H, Muller-Hill B. The three operators of the lac operon cooperate in repression. *EMBO J*. 1990;9(4):973–9.
13. Jacob C, Barbasiewicz A, Tsui G. Swarms and genes: Exploring λ - switch gene regulation through swarm intelligence. In: 2006 IEEE International Conference on Evolutionary Computation. Vancouver, BC, Canada: IEEE; 2006. p. 2535–542.
14. Coumans E. *Bullet physics library*. Open source: bulletphysics.org. 2013.
15. Jacob C, Burleigh I. Genetic programming inside a cell. *Genet Program Theory Pract III*. 2006;9:191–206.
16. Loew LM, Schaff JC. The Virtual Cell: a software environment for computational cell biology. *TRENDS in Biotechnology*. 2001;19(10):401–406.
17. Andrews SS, Addy NJ, Brent R, Arkin AP. Detailed simulations of cell biology with Smoldyn 2.1. *PLoS Comput Biol*. 2010;6(3):e1000705.
18. Vilar JMG, Guet CC, Leibler S. Modeling network dynamics: the lac operon, a case study. *J Cell Biol*. 2003;161(3):471.
19. Novick A, Weiner M. Enzyme Induction as an All-or-None Phenomenon. *Proc Natl Acad Sci U S A*. 1957;43(7):553–66.
20. Santillan MES, Mackey MC. Quantitative approaches to the study of bistability in the lac operon of *Escherichia coli*. *J R Soc Interface R Soc*. 2008;5 Suppl 1:29–39.
21. Wong P, Gladney S, Keasling JD. Mathematical model of the lac operon: inducer exclusion, catabolite repression, and diauxic growth on glucose and lactose. *Biotechnol Prog*. 1997;13(2):132–43.
22. Yildirim N, Mackey MC. Feedback regulation in the lactose operon: A mathematical modeling study and comparison with experimental data. *Biophys J*. 2003;84(5):2841–51.
23. Santillan MES, Mackey MC. Influence of catabolite repression and inducer exclusion on the bistable behavior of the lac operon. *Biophys J*. 2004;86(3):1282–92.
24. van Hoek MJA, Hogeweg P. In silico evolved lac operons exhibit bistability for artificial inducers, but not for lactose. *Biophys J*. 2006;91(8):2833–43.
25. van Hoek M, Hogeweg P. The effect of stochasticity on the lac operon: an evolutionary perspective. *PLOS Comput Biol*. 2007;3(6):111.
26. Ozbudak EM, Thattai M, Lim HN, Shraiman BI, Van Oudenaarden A. Multistability in the lactose utilization network of *Escherichia coli*. *Nature*. 2004;427(6976):737–40.
27. Santillan M. Bistable behavior in a model of the lac operon in *Escherichia coli* with variable growth rate. *Biophys J*. 2008;94(6):2065–81.
28. Gillespie DT. Approximate accelerated stochastic simulation of chemically reacting systems. *J Chem Phys*. 2001;115(4):1716–33.
29. Gillespie DT, Petzold LR. Improved leap-size selection for accelerated stochastic simulation. *J Chem Phys*. 2003;119(16):8229–34.
30. Galassi M, Gough B. *GNU Scientific Library Reference Manual*. Bristol, UK: Network Theory Ltd; 2006.
31. Abbott RG, Forrest S, Pienta KJ. Simulating the hallmarks of cancer. *Artif Life*. 2006;12(4):617–34.
32. Casal A, Sumen C, Reddy TE, Alber MS, Lee PP. Agent-based modeling of the context dependency in T cell recognition. *J Theor Biol*. 2005;236(4):376–91.
33. An G. In silico experiments of existing and hypothetical cytokine-directed clinical trials using agent-based modeling. *Crit Care Med*. 2004;32(10):2050–60.
34. Peirce SM, Van Gieson EJ, Skalak TC. Multicellular simulation predicts microvascular patterning and in silico tissue assembly. *FASEB J: Off Publ Fed Am Soc Exp Biol*. 2004;18(6):731–3.
35. Longo D, Peirce SM, Skalak TC, Davidson L, Marsden M, Dzamba B, DeSimone DW. Multicellular computer simulation of morphogenesis: Blastocoel roof thinning and matrix assembly in *Xenopus laevis*. *Dev Biol*. 2004;271(1):210–22.
36. Bonabeau E. Agent-based modeling: methods and techniques for simulating human systems. *Proc Natl Acad Sci U S A*. 2002;99 Suppl 3(90003):7280–7.
37. Jacob C, Burleigh I. Biomolecular swarms: an agent-based model of the lactose operon. *Nat Comput*. 2004;3(4):361–76.
38. Jacob C, von Mammen S, Davison T, Sarraf-Shirazi A, Sarpe V, Esmaili A, et al. Lindsay virtual human: Multi-scale, agent-based, and interactive. In: Kotodziej J, Khan S, Burczynski T, editors. *Advances in Intelligent Modelling and Simulation: Artificial Intelligence-based Models and Techniques in Scalable Computing*. Berlin, Germany: Springer; 2012. p. 327–49.

39. Junker G. Pro OGRE 3D Programming: Apress; 2006.
40. Sarpe V, Jacob C. Simulating the decentralized processes of the human immune system in a virtual anatomy model. *BMC Bioinformatics*. 2013;14(Suppl 6):1–26.
41. Sarraf Shirazi A, von Mammen S, Jacob C. Abstraction of Agent Interaction Processes. *Simul: Trans Soc Mode Simul Int*. 2013;89(4):524–38.
42. Mammen S, Phillips D, Davison T, Jamniczky H, Hallgrímsson B, Jacob C. Swarm-Based Computational Development In: Doursat R, sayama H, Michel O, editors. *Morphogenetic Engineering: Understanding Complex Systems*. Berlin, Heidelberg: Springer; 2012. p. 473–99.
43. Berman HM. The Protein Data Bank. *Nucleic Acids Res*. 2000;28(1):235–42.
44. Karp PD, Riley M, Paley SM, Pellegrini-Toole A, Krummenacker M. EcoCyc: Encyclopedia of *Escherichia coli* genes and metabolism. *Nucleic Acids Research*. 1998;26(1):50–53.
45. Mirkin EV, Mirkin SM. Mechanisms of transcription-replication collisions in bacteria. *Mol Cell Biol*. 2005;25(3):888–95.
46. Grunberg-Manago M. Messenger RNA stability and its role in control of gene expression in bacteria and phages. *Annu Rev Genet*. 1999;33:193–227.
47. Pedersen M, Nissen S, Mitarai N, Lo Svenningsen S, Sneppen K, Pedersen S. The functional half-life of an mRNA depends on the ribosome spacing in an early coding region. *J Mol Biol*. 2011;407(1):35–44.
48. Young R, Bremer H. Polypeptide-chain-elongation rate in *Escherichia coli* B/r as a function of growth rate. *Biochem J*. 1976;160(2):185–94.
49. Zygote Media Group Inc. 3D Science. 2011. <http://www.zygote.com>. Accessed Sept 2015.
50. Sarraf Shirazi A, Davison T, von Mammen S, Denzinger J, Jacob C. Adaptive agent abstractions to speed up spatial agent-based simulations. *Simul Model Pract Theory*. 2014;40:144–60.
51. Reeves WT. Particle systems—a technique for modeling a class of fuzzy objects. *ACM Trans Graphics*. 1983;2(2):91–108.
52. Franz B, Erban R. In: Lewis MA, Maini PK, Petrovskii SV, editors. *Hybrid Modelling of Individual Movement and Collective Behaviour*. Lecture Notes in Mathematics, vol 2071. Berlin, Germany: Springer; 2013, pp. 129–57.
53. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat T, Weissig H, et al. *Nucleic Acids Res*. 2000;28(1):235–42.

**Submit your next manuscript to BioMed Central
and take full advantage of:**

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at
www.biomedcentral.com/submit

