



AgentCell: A Digital Single-Cell Assay for Bacterial Chemotaxis

Thierry Emonet^{1*}, Charles M. Macal², Michael J. North², Charles E. Wickersham¹, Philippe Cluzel¹

¹ The Institute for Biophysical Dynamics and the James Franck Institute, The University of Chicago, 5640 S. Ellis Av., Chicago, IL 60637.

² Center for Complex Adaptive Agent Systems Simulation, Decision and Information Sciences Division, Argonne National Laboratory, 9700 S. Cass Ave., Argonne IL 60439.

ABSTRACT

Motivation: In recent years, single-cell biology has focused on the relationship between the stochastic nature of molecular interactions and variability of cellular behavior. To describe this relationship, it is necessary to develop new computational approaches at the single cell level.

Results: We have developed AgentCell, a model using agent-based technology to study the relationship between stochastic intracellular processes and behavior of individual cells. As a test-bed for our approach we use bacterial chemotaxis, one of the best-characterized biological systems. In this model, each bacterium is an agent equipped with its own chemotaxis network, motors and flagella. Swimming cells are free to move in a 3D environment. Digital chemotaxis assays reproduce experimental data obtained from both single cells and bacterial populations.

Availability: AgentCell is available on request from the authors.

Contact: emonet@uchicago.edu

Supplementary information: available on the OUP server.

1 INTRODUCTION

In a recent experimental study, Korobkova et al. (2004) showed that behavioral variability of an individual cell could be the result of the stochastic nature of molecular interactions in intracellular signaling pathways. Consequently, even genetically identical cells can exhibit different behaviors (Spudich and Koshland, 1976). These studies suggest that stochastic molecular events in signaling pathways play a significant role in single cell behavior. In this paper, we present AgentCell, a digital assay designed to study how molecular noise influences the behavior of a swimming single cell in a 3D environment.

Most computational models characterize biological systems at one specific scale of interest: e.g. molecular, cellu-

lar, or inter-cellular. Computational tools such as GEPASI (Mendes 1993), DBSolve (Goryanin et al., 1999) and StochSim (Morton-Firth 1998, Le Novère and Shimizu 2001) model intracellular biochemical reactions within one cell. At the whole cell level, there exist more integrative tools including E-Cell (Tomita et al. 1999) and Virtual Cell (Loew and Schaff, 2001). Some of these tools include capabilities to handle multiple timescales (Takahashi et al. 2004). Cell populations have been modeled with partial differential equations, gas kinetic theory, cellular automata and Brownian agents (e.g. Chen et al. 2003, Schweitzer 2003, Leawus and Ford 2001, Schnitzer 1993, Ford and Lauffenburger 1991). A recent software framework for the simulation of morphogenesis, CompuCell, uses a combination of these techniques (Izaguirre et al. 2004).

We developed AgentCell, an agent-based model for *simultaneously* modeling biochemical processes within individual cells and the associated motion of cells within a 3D environment. The outer layer of our model is the agent-based simulation framework Repast, a Java-based software toolkit that has been routinely used to model socio-economic systems (Collier and Sallach, 2001). The long-term goal of this work is to develop an agent-based software to describe population dynamics from autonomous interacting bacteria that can make decisions independently from each other. The representation of biological entities as *autonomous* agents competing in a common environment is particularly suited to model the evolutionary aspects of biological organization (see Section 2.1).

As a test-bed for our computational approach we simulated the chemotactic behavior of *Escherichia coli* bacteria. Bacterial chemotaxis in *E. coli* is one of the best characterized examples of information processing in biology (Berg 2000). Environmental signals are converted into molecular intracellular information via a signal transduction network. *E. coli* is able to swim up or down chemical gradients using flagella as propellers (Supplementary Figure 1). Each flagel-

* To whom correspondence should be addressed.

lum rotates under the action of a rotary motor. When most of the motors rotate counterclockwise (CCW), the flagella form a bundle and the bacterium swims smoothly. When motors rotate clockwise (CW), the bacterium tumbles erratically. Tumbles randomize the cell trajectory, and their temporal modulation allows bacteria to perform chemotaxis by swimming toward attractants or away from repellents. The chemotaxis network in *E. coli* senses temporal changes in environmental chemical concentrations and transduces external chemical signals into intracellular information. In a simple picture of the chemotaxis network, the external concentration of ligand is the network input. The internal concentration of signaling molecules (CheY-P) is the network output. CheY-P binds preferentially to the motor, and the CW bias of the flagellar motors, that is, the fraction of time that a single motor spends rotating in the CW direction, increases with CheY-P concentration (Cluzel et al. 2000). There exist several excellent molecular and physiological reviews of signaling networks (Sauro and Kholodenko 2004) and bacterial chemotaxis (Berg 2000).

We define single bacteria as agents whose goal is to swim upwards gradients of chemical nutrients. Within one cell, receptor complexes provide the interface between the bacterium and the local environment that can contain concentration fields of chemical effectors. Biochemical networks are currently implemented using instances of the stochastic simulator StochSim (Morton-Firth 1998).

To validate our computational approach we simulated the chemotactic response to a linear gradient of attractant of individual swimming bacteria. At regular time intervals, we recorded the position and orientation of the cell as well as the state of the motor, flagella, receptors and the activity of each protein involved in the chemotaxis pathway. We compare our simulations to experimental data obtained from both single cells and population measurements.

2 SYSTEMS AND METHODS

2.1 Agent-based Approach

Agent-based modeling draws on several fields including computer science, artificial intelligence, complex systems, and the social sciences. Agent-based simulations have been developed to model the sociology of human societies (Epstein and Axtel 1976), competition and collaboration (Axelrod 1997), the spread of epidemics (Gordon 2003), and the organizational behavior of canid populations (Pitt et al. 2003).

Resnick (1994) provides a general introduction to agent-based computing. Jennings (2000) identifies the essential characteristics of agents and compares agent-oriented software engineering with object-oriented programming. According to his definition, agents are “problem solving entities with well-defined boundaries and interfaces”. Agents have goals and can determine if their situation becomes bet-

ter or worse relative to the fulfillment of these goals. Agents act on locally available information. Global or system-wide information is not accessible.

The main difference between agents and objects is that agents are autonomous: they have the ability to control their internal state and behavior without the direction of a central authority (Wooldridge 1997). Autonomy decentralizes decision making and therefore greatly simplifies the implementation of the whole system’s control. Because each agent decides by itself when to act and what action to perform, there is no need for a complex centralized decision making entity. Finally, agents follow protocols to interact with their environment and to interact with each other. Because of their autonomy, agents are free to make *run-time decisions* about the scope (with whom to interact) and nature of the interactions. Flexibility in the timing, scope and nature of the interactions is one of the strengths of agent-based programming (Jennings 2000).

The modularity of agent-based approaches is very suited to study the natural architecture of biological systems (Hartwell et al. 1999). The flexibility of the interactions between agents facilitates the software representation of the interactions between modules within biological systems. Finally, the autonomy of individual agents simplifies the implementation of phenotypic variability within a population of cells (Spudich and Koshland 1976). In particular, it allows us to model the intracellular pathways inside the individual agent (cell) with a stochastic simulation. Thus, the cell behavior directly reflects its intracellular stochastic fluctuations.

In this first version of AgentCell, we used only some of the features offered by the agent-based representation. Cells are the sole agents while cellular organelles (motors, receptors, etc.) are simple objects. These objects have no autonomy, and are controlled by the cell. Redefining cellular organelles (e.g. motors) as agents can be implemented in future versions of AgentCell. Although we did not include any interactions between cells, the strength of the agent-based approach in modeling agent-to-agent interactions will be essential for our long term goal. Allowing cell-to-cell interactions requires message passing between agents (see Sections 4, 6). This first version provides several scripts to run simultaneously thousands of realizations of non-interacting single cells on massively parallel computers. We used this feature to approximate the behavior of bacterial populations with averages over the realizations of independent non-interacting cells (Section 5).

2.2 Architecture

Figure 1 shows the architecture of AgentCell. The main packages and classes in this code are:

Models: This package contains classes that control different type of simulations. The main controller for the simulations presented in this paper is *ChemotaxisModel*.

World: This class defines the space in which the cells (agents) live. *World* contains a *Collection* of cells as well as an instance of the *BoundaryConditions* class, essentially a *Collection* of boundaries that can be either reflective or periodic. For simplicity, reflective boundaries reflect oriented positions as if they were particles bouncing elastically against a wall.

Cells: The default attributes of a cell are position, orientation, volume and *motionStepper* (see below). They are provided by the abstract class *Cell*. Each instance of the *Cell* class is an independent agent. The *ChemotaxisCell* class extends *Cell* with attributes necessary to perform chemotaxis: *Receptors*, *Network*, *Motor*, *Flagella*.

Receptors: This class provides a method to read the concentration of ligand from the local environment.

Network: All biochemical networks use the *Network* interface. This interface declares methods to set and get the number of molecules in the networks. It also declares a *step* method to advance the network in time. *Network* is implemented by the class *Stochsim*, a wrapper around the StochSim software package. *Stochsim* uses the same input files as the original StochSim program to define the topology and reactions rates of the network. For the chemotaxis pathway, this is done in the *ChemotaxisNetwork* class. New additional pathways require a separate set of Stochsim input files.

Motor: Bacterial motors rotate either clockwise (CW) or counterclockwise (CCW). The abstract class *Motor* keeps track of the binary state (CW or CCW) of the effective motor and provides a method to switch from one state to the other. It also declares an abstract *step* method to advance the state of the motor in time. The abstract class *ThresholdMotor* extends *Motor* with a time step method that switches the motor state from CCW to CW if a threshold condition is satisfied. The threshold condition is defined in subclasses: *AveragedCheYpThresholdMotor* defines the threshold using the recent history of the intracellular level of CheY-P.

Flagella: Changes in the states of the motors do not always translate into changes in swimming behavior. The abstract class *Flagella* handles this discrepancy. The current model allows two possible states for the flagella: *bundled* when the flagella form a corkscrew bundle and *apart* when the flagella fly apart. The subclasses of *Flagella*, *TetheredFlagellum* and *SwimmingFlagella*,

switch the state of the flagella according to the state of the motor.

Motion: This package controls the motion of a cell within the *World*. Two interfaces, *Motion* and *MotionStepper* handle the cellular motion. *Motion* represents cell movements, like a run or a tumble. *MotionSteppers* calls the *step* routines of the different types of movement. For chemotaxis, *Motion* is implemented by the *Run* and *Tumble* classes. *MotionStepper* is implemented by the *RunTumbleStepper* class which is responsible for executing a *Run* or a *Tumble* step depending on the current state of the *Motor*.

Molecules: This package defines molecular types (*Molecule*), concentration (*Concentration*), copy numbers (*CopyNumber*), and concentration fields (*ConcentrationField*).

2.3 Time Stepping

In agent-based simulations, the scheduler is a piece of software responsible for stepping the system through time. Scheduling consists of keeping a global clock, updating the clock to the next event, and maintaining a sorted list of events. Each agent inserts its own future events inside the scheduler's list of events (Supplementary Information).

In the simulations presented below, the agents (cells) update their internal state and position every 0.01 seconds. This time interval is the resolution used to measure the molecular noise in individual bacteria (Korobkova et al. 2004). Different parts of a cell typically evolve on very different dynamical time scales. For example, the timescale of the fastest reactions can be several orders of magnitude smaller than the timescales involved in the motion of the bacteria. For this reason, when the cell executes one step, the network makes several smaller internal sub-steps (approximately 50,000 sub-steps for the chemotaxis model used in Section 5).

This simple time stepping strategy is sufficient for the simulations in this paper. However, the agent-based approach (in particular Repast) permits much more complex scheduling with, for example, causal relationships between events.

3 ALGORITHMS FOR CHEMOTAXIS

3.1 Chemotaxis Network

The chemotaxis network was simulated using the same model as in Morton-Firth, Shimizu and Bray (1999) and Korobkova et al. (2004). The input of the network is the receptor occupancy (the probability for a receptor to be ligand bound). The output is the number of molecules of response regulator CheY-P inside the cell.

3.2 Receptor binding

We assumed that ligand binding operates at quasi-equilibrium and that the ligand-receptor affinity depends on the activity state of the receptor (see e.g. Morton-Firth et al. 1999). Thus, the probability q for an inactive receptor to be ligand bound was $q = [L]/([L]+K_D)$ with $[L]$ the concentration of ligand and K_D the dissociation constant. For active receptors the formula remains identical but the value of K_D is smaller (see Table 1).

There exist more sophisticated models of the receptors, in particular models that enhance the gain of the system by adding coupling between different chemoreceptor species (Mello and Tu 2003) and conformational spread within receptor clusters (Shimizu et al., 2003). Because of the modularity of AgentCell, these models can be readily included. In fact, one of these models (Shimizu et al. 2003) is already available in AgentCell through the *Stochsim* class.

3.3 Motor switch

The CW bias or probability for the motor to rotate clockwise is very sensitive to the concentration of CheY-P inside the cell. Measurements in single cells have revealed a non-linear motor behavior with CheY-P characterized by a Hill coefficient of 10 (Cluzel et al. 2000). To convert the output of the chemotaxis network into motor states, we use a simple threshold model with memory and a filter that eliminate short events (class *AveragedCheYpThresholdMotor*).

The algorithm is the following. The time average of CheY-P over the last 0.3 seconds (see Table 1) is compared with the threshold level of CheY-P (defined at initialization time). If the threshold is crossed, the binary state of the motor is switched. Experimental measurements in single cells (Figure 2 and Supplementary Figure 3) were reproduced by adjusting the threshold level and the size of the averaging window (0.3 seconds). The user can specify a lower limit for the time spent in either state. The motor is not allowed to switch unless it has spent the required minimum amount of time in one state.

3.4 Flagella state

In *TetheredFlagellum*, the state of the flagellum exactly reproduces the state of the motor with CW→*apart* and CCW→*bundled*. In *SwimmingFlagella*, we assume that in 20% of the cases, a switch of the motor state from CCW to CW fails to translate into a switch of the flagella state from *bundled* to *apart*. This assumption is based on the relation between the CW bias and [CheY-P] as measured in single swimming cells (data not shown).

3.5 Motion: run

In line with Berg and Brown (1972) who tracked individual cells in 3D, we assumed that during a run the speed of the cell is constant and that the direction of motion is affected by rotational Brownian diffusion. The position and orientation of the cell are updated in the *step* method of the class

Run. The time integration of position and orientation is described in the Supplementary Information.

3.6 Motion: tumble

During a tumble (CW rotation), the position of the cell is held constant. The new direction of the motion is chosen randomly from one of the directions forming an angle α with the original direction of motion. The angle α is drawn from a Gamma distribution with shape parameter 4, scale parameter 18.32 and location parameter -4.6. With these values the resulting distribution of tumble angles fits the distribution of tumbling angles measured by Berg and Brown (Figure 3, 1972).

4 SOFTWARE

We built AgentCell using the agent-based Repast framework (Collier and Sallach, 2001). Repast is an agent-based framework similar to its predecessor Swarm (Burkhart et al., 2000), but differs in several important respects. Owing to its implementation in Java, Repast is platform independent, web-compatible, and allows integration across a variety of languages. All the capabilities of Repast are directly available within AgentCell. In particular, Repast provides the software apparatus for creating, running, displaying and collecting data from an agent-based simulation. It also has libraries for handling complex interactions between agents. In this first version of AgentCell, we only used a few features from Repast. The *ChemotaxisModel* class extends the *SimpleModel* class from Repast. It is responsible for starting and terminating higher-level housekeeping tasks, such as I/O, random number generation, threaded process assignment, interfacing with the graphical user interface, and scheduling.

In addition to Repast, AgentCell uses the Apache Log4J high performance Java-based results logging system (Gülcü 2003), the National Center for Supercomputing Applications' (NCSA) *Hierarchical Data Format 5 (HDF5)* (NCSA 2004) data storage system, and the Colt mathematics library among other tools (Hosccek 2004). These libraries provide AgentCell with useful functionalities such as the ability to store results in either HDF5 or in comma separated value (CSV) format.

The AgentCell project is focused on developing an agent-based model of bacterial chemotaxis to simulate bacterial populations on massively parallel computers rather than general software for the PC. As such, AgentCell does not currently have an interactive user interface. Instead, users can specify the required cellular configuration, world structure, and parameters using Java code in tools such as Eclipse (Eclipse, 2004) (see Supplementary Information for details on how to use AgentCell). The AgentCell outputs have been analyzed with IDL (Research Systems 2004) and Mathematica (Wolfram Research 2004).

Probabilistic aspects of intracellular signaling are important. For this reason, we use a stochastic approach for the time integration of the biochemical networks. We implemented the *Network* interface with the class *Stochsim*, a wrapper around the general purpose biochemical simulator StochSim (Morton-Firth 1998). StochSim represents individual molecules or molecular complexes as individual software objects. StochSim has been used extensively to simulate the chemotaxis network in *E. coli* (See Shimizu et al. 2003 and references therein). Recently, some of us used StochSim to study the relationship between molecular noise and behavior in a single *E. coli* bacterium (Korobkova et al. 2004). Implementing the *Network* class with StochSim required accessing the internal time loop of StochSim in order to feed or extract information at the time intervals required by the agent-based simulation. For this purpose we modified StochSim to appropriately pause and then continue as required by Repast.

StochSim runs efficiently within Repast. A StochSim (version 1.4) simulation of the chemotaxis network without stimulus runs 1000 simulated seconds in 53 minutes on one processor of a Linux (Fedora core 3) computer with dual 2.40 GHz Intel Xeon processors and 2 GB of RAM. AgentCell with one cell in a 3D environment but without receptors ran in 58 minutes on the same computer. AgentCell with one cell in a 3D environment with a ligand gradient ran in 83 minutes on the same computer.

To simulate cell populations, individual cells must run on separate processors. In the current version of AgentCell, cells cannot exchange information. Allowing cell-to-cell interactions implies parallelizing the code for its use over distributed systems. This feature will be developed in the next version of AgentCell. However, the current version of AgentCell can use distributed computing clusters to execute simultaneously independent realizations of a single cell evolution. For each realization, the input files are less than 2 MB combined, and the output files can be a few hundred MB. The only conflict for shared cluster resources is for the network and file system when simulation results are saved. Thus, AgentCell can achieve nearly linear speedup for clusters with efficient file systems (Supplementary Information). Adding cell-to-cell interactions to the model will most likely cause AgentCell to exhibit less than linear speedup. Simulations with only one or two cells were also run on laptops and desktops both under Linux or Windows.

5 MODEL VALIDATION

To validate our approach, we simulated chemotactic assays and compared them with published experimental data. Topology, reaction rates and protein concentrations of the chemotaxis network were chosen exactly identical to those used in Morton-Firth et al. (1999) and Korobkova et al. (2004). Additional parameters defining the cells are listed in

Table 1. Cells were put in a 3D environment. At time $t=0$, the initial position, motor state, flagella state and internal protein levels of the chemotaxis network were identical for all the cells. Only orientation and random generator seeds were shuffled. Cellular behavior was simulated over 40 minutes. The internal state of each cell (position, orientation, internal level of CheY-P, state of the motors, state of the flagella, etc.) was recorded every 0.01 seconds.

Table 1. Parameters used in the chemotaxis model.

Parameter	Symbol	Value
Internal cell volume	V	1.41×10^{-15} L
Cell radius	a	1 μ m
Constant run speed	V	20 μ m sec $^{-1}$
Viscosity	η	0.027 g(cm sec) $^{-1}$
Temperature	T	305 °K
Rotational diffusion	$D_r = kT / (8\pi \eta a^3)$	0.06205 sec $^{-1}$
CheY-P threshold		$\langle Y_p \rangle + 0.55 \sigma(Y_p)$
Motor CheY-P averaging window		0.3 sec
Minimum CW interval		0.1 sec
Minimum CCW interval		0.1 sec
Receptor-aspartate dissociation constant (inactive receptors)	K_D	1.71×10^{-6} M
Receptor-aspartate dissociation constant (active receptors)	K_D	12×10^{-6} M

$\langle Y_p \rangle$ and $\sigma(Y_p)$ represent the mean and standard deviation of CheY-P.

We performed two simulations. In the first simulation (hereafter simulation 1), we tested the behavior of a population of 1166 unstimulated cells. The cells were placed in a 3D environment that was infinite in all directions and free of attractant. We compare the results from simulation 1 with two experimental assays: tethered and swimming cells. Although the digital cells in simulation 1 are swimming, the state of the motors (instead of the state of the flagella) can be compared with the state of the motors in tethered cell experiments.

The second simulation (hereafter simulation 2) was designed to test the chemotactic response from digital cells. The world was a half infinite space $z \geq 0$ with a reflecting boundary at $z=0$. 540 cells started to swim from the initial position z_0 located 20 μ m away from the boundary. At time $t=0$, the constant gradient of aspartate $10^{-8} \nabla z$ M/ μ m was established.

5.1 Single cell behavior

We compare our results with recently published (Korobkova et al. 2004) distributions of CW and CCW time intervals obtained by monitoring the switching events of individual flagellar motors from non-stimulated wild type cells (Figure 2a). We plot in Figure 2b the distributions resulting from the binary time series of the motor states in one cell of the digital population from simulation 1. We chose a digital cell with approximately the same averaged CW bias as for the

real cell. The CW bias is the total time spent in CW divided by the total length of the time series. The simulation data accurately reproduces the important statistical features observed in single cells, such as the long tail deviation from an exponential distribution of the CCW intervals. Our single cell model also reproduces the main spectral features from the binary time series of motor states measured by Korobkova et al. (2004) in single wild type cells (Supplementary Figure 3).

Figure 3 shows the chemotactic response of a single digital cell (simulation 2). It illustrates how AgentCell can help relate the stochastic time evolution of the intracellular amount of CheY-P (panel **a**) to the trajectory of the cell along the aspartate gradient (panel **b**). Whenever the cell moves upwards, its receptors bind ligand. The response of the chemotaxis network consists of decreasing the amount of CheY-P, which causes the cell to run. Because of the rotational diffusion, however, the cell loses track of the original direction and some times ends up running downwards the gradient (e.g. between 50 and 100 seconds). Under this latter condition, the chemotaxis network increases the tumbling rate, and a new random direction is chosen.

5.2 Population behavior

In the absence of attractant, the concentration of *E. coli* satisfies the scalar diffusion equation. Thus, the population average of the square distance to the origin follows Einstein's relation $\langle r^2 \rangle = 6 D t$, where D is the macroscopic random motility coefficient (e.g. Berg 1993). Taking the average of r^2 over the 1166 cells of simulation 1 we obtained $D = 4.21 \times 10^{-6} \text{ cm}^2/\text{s}$ (Supplementary Information). Leawus and Ford (2001) measured $D = 3.8 \pm 0.2 \times 10^{-6} \text{ cm}^2/\text{s}$; Berg (1993, p. 93) reports $D = 4 \times 10^{-6} \text{ cm}^2/\text{s}$.

The chemotactic response of a population of digital cells is illustrated in Figure 4. We plot, as function of time, the number of cells from simulation 2 within the region defined by $z > 1.2 \text{ mm}$. We also simulated cells diffusing into the same region in absence of attractant. We define the chemotactic response as the ratio of bacteria accumulated in the region of interest in the presence of attractant to accumulation in its absence. After about 400 seconds the receptors of the stimulated cells saturate and the corresponding behavior becomes purely diffusive. Before saturation the chemotaxis response is between 4 and 5. This numerical result reproduces qualitatively the response (about 15) measured from high-throughput capillary assays (Bainer et al. 2003).

6 CONCLUSION

We have developed AgentCell, a model that uses agent-based technology to study the effect of intracellular stochastic fluctuations on the behavior of single cells. We tested our approach by modeling the chemotactic response of single *E. coli* cells to chemo-attractant gradients in a 3D environment.

Our data reproduce several key aspects of experiments from both single cells and populations measurements.

Our long-term goal is to develop a tool to study the relationship between intracellular computation and population behavior all within one computational framework. The current version of AgentCell is the first step in this direction. The representation of cells as agents will allow us to introduce complex interactions between cells in the next version of AgentCell. Agent-based features could be used to study cell-to-cell communication such as quorum sensing and hydrodynamic coupling, which relate intracellular signaling to population behavior (Shapiro 1998).

ACKNOWLEDGEMENTS

This work is partially supported by joint research funding under H.28 of the U. S. Department of Energy Contract W-31-109-ENG-38. The authors are thankful to R. Rosner for suggesting the use of the agent based approach and to T. Shimizu and C. Guet for comments on the manuscript.

REFERENCES

- Axelrod, R. (1997) *The Complexity of Cooperation: Agent-based Models of Competition and Collaboration*. Princeton Univ. Press, Princeton, NJ
- Bainer, R., Park, H., and Cluzel, P. (2003) A high-throughput capillary assay for bacterial chemotaxis. *J. of Microbiol. Meth.*, **55**, 315-319.
- Berg, H.C. and Brown D.A. (1972) Chemotaxis in *Escherichia coli* analyzed by Three-dimensional Tracking. *Nature*, **239**, 500-504
- Berg, H.C., (1993), *Random Walks in Biology*. Princeton Univ. Press, Princeton, NJ
- Berg, H.C. (2000) Motile behavior of bacteria. *Physics Today*, **53**, 24-29
- Burkhardt, R.M., Ashkenazi, M. and Minar, N. (2000) *Swarm Release Documentation*. <http://www.santafe.edu/projects/swarm/swarmdocs/set/set.html>.
- Chen, K.C., Ford, R.M., Cummings, P.T. (2003) Cell balance equation for chemotactic bacteria with a biphasic tumbling frequency. *J. Math. Biol.*, **47**, 518-546
- Cluzel, P., Surette, M. and Leibler, S. (2000) An ultrasensitive bacterial motor revealed by monitoring signaling proteins in single cells. *Science*, **287**, 1652-1655
- Collier, N. and Sallach D. (2001) *Repast*, <http://repast.sourceforge.net>
- Eclipse Foundation (2004) <http://www.eclipse.org/>.
- Epstein, J.M. and Axtell, R. (1996) *Growing Artificial Societies: Social Science from the Bottom Up*. MIT Press, Cambridge, MA
- Ford, R.M., Lauffenburger, D.A. (1991) Analysis of chemotactic bacterial distributions in population migration assays using a mathematical model applicable to steep or shallow attractant gradients. *Bull. Math. Biol.*, **53**, 721-749
- Gordon, T.J. (2003). A simple model of an epidemic. *Technol. Forecast. Soc. Change*, **70**, 397-417

- Goryanin, I., Hodgman, T.C., and Selkov, E. (1999) Mathematical Simulation and Analysis of Cellular Metabolism and Regulation. *Bioinformatics*, **15**, 749–758
- Gülcü, C. (2003), *The Complete log4j Manual*. QOS.ch, Lausanne, Switzerland
- Hartwell L.H., Hopfield J.J., Leibler S. and Murray A.W. (1999) From molecular to modular cell biology. *Nature*, **402**, c47-c52
- Hoschek, W. (2004) *Colt Home Page*. <http://hoschek.home.cern.ch/hoschek/colt/>
- Izaguirre, J.A., Chaturvedi, R., Huang, C., Cickovski, T., Coffland, J., Thomas, G., Forgacs, G., Alber, M., Hentschel, G., Newman, S.A. and Glazier J.A. (2004) CompuCell, a multi-model framework for simulation of morphogenesis. *Bioinformatics*, **20**, 1129-1137
- Jennings, Nicholis R. (2000) On agent-based software engineering. *Artificial Intelligence*, **117**, 277-296
- Korobkova, E., Emonet, T., Vilar, J. M. G., Shimizu, T.S., Cluzel, P. (2004) From molecular noise to behavioural variability in a single bacterium. *Nature*, **428**, 574–578
- Le Novere, N. and Shimizu, T.S. (2001) Stochsim: Modeling of Stochastic Biomolecular Processes. *Bioinformatics*, **6**, 575–576
- Leawus P. and Ford R.M. (2001) Quantification of Random Motility and Chemotaxis Bacterial Transport Coefficients Using Individual-Cell and Population-Scale Assays. *Biotech. and Bioengin.*, **75**, 292-304
- Melo, B.A. and Tu, Y. (2003), Quantitative Modeling of sensitivity in bacterial chemotaxis: The role of coupling among different chemoreceptor species. *Proc. Natl. Acad. Sci. U S A*, **100**, 8223-8228
- Mendes, P. (1993) Gepasi: A Software Package for Modeling the Dynamics, Steady States, and Control of Biochemical and Other Systems. *Comput. Appl. Biosci.*, **9**, 563–571
- Morton-Firth, C.J. (1998) *Stochastic Simulation of Cell Signaling Pathways*. PhD Thesis, University of Cambridge, Cambridge, UK
- Morton-Firth, C.J., Shimizu, T.S., and Bray, D. (1999) A Free-Energy-Based Stochastic Simulation of the Tar Receptor Complex. *J. Mol. Biol.*, **286**, 1059–1074.
- NCSA, (2004), *HDF 5*, <http://hdf.ncsa.uiuc.edu/HDF5/>
- Pitt, W.C., Box, P.W. and Knowlton, F.F. (2003) An individual-based model of canid populations: modeling territoriality and social structure. *Ecological Modeling*, **166**, 109-121.
- Research Systems, Inc. (2004), <http://www.rsinc.com/idl/>
- Resnick, M. (1994) *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*. MIT Press, Cambridge, MA
- Sauro H.M. and Kholodenko B.N. (2004) Quantitative analysis of signaling networks. *Prog. Biophys. Mol. Biol.*, **86**, 5-43
- Schnitzer, M.J. (1993) Theory of continuum random walks and application to chemotaxis. *Phys. Rev. Lett. E*, **48**, 2553-2568
- Schweitzer, F (2003) *Brownian Agents and Active Particles. Collective Dynamics in the Natural and Social Sciences*. Springer, Berlin, Germany
- Shapiro, J.A. (1998) Thinking about bacterial populations as multicellular organisms. *Annu. Rev. Microbiol.*, **52**, 81-104
- Shimizu TS, Aksenov SV and Bray D. (2003) A spatially extended stochastic model of the bacterial chemotaxis signaling pathway. *J. Mol. Biol.*, **329**, 291-309
- Spudich, J.L. and Koshland D.E. Jr. (1976) Non-genetic individuality: chance in the single cell. *Nature*, **262**, 467-471
- Takahashi K., Kaizu K., Hu B., and Tomita M. (2004) A multi-algorithm, multi-timescale method for cell simulation, *Bioinformatics*, **20**, 538–546
- Tomita M., Hashimoto K., Tkahashi K., Shimizu, T.S., Matsuzaki, Y., Miyoshi, F., Saito, K., Tanida, S., Yugi, K., Venter, C. and Hutchison III, C.A. (1999) E-Cell: Software Environment for Whole Cell Simulation, *Bioinformatics*, **15**, 72–84.
- Loew L.M. and Schaff, J.C., (2001) The Virtual Cell: A Software Environment for Computational Cell Biology. *Trends in Biotechnology*, **19**, 401–406.
- Wolfram Research, Inc., (2004) <http://www.wolfram.com/>
- Wooldridge, M. (1997) Agent-based software engineering. *IEE Proc. Software Engineering*, **144**, 26-37

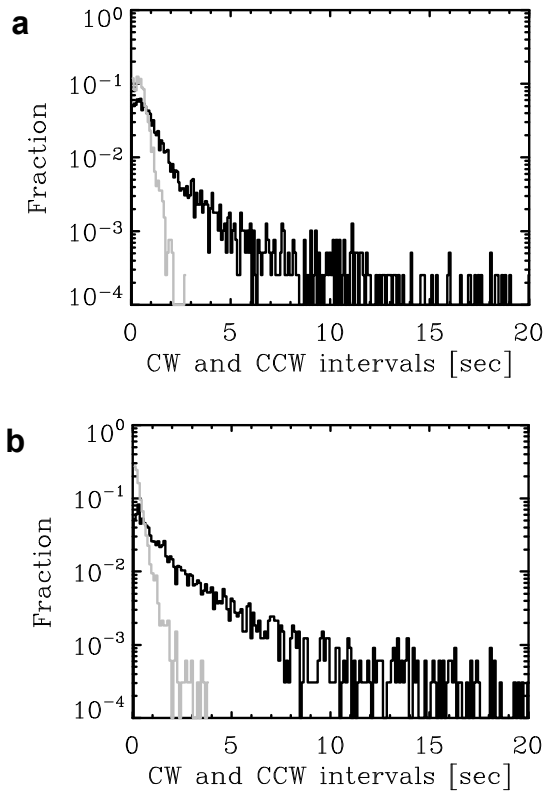


Fig. 2. Fraction of CW and CCW intervals of individual flagellar motors from non-stimulated wild-type cells in a medium without attractant. (a) real cell (Korobkova et al. 2004, Fig. 2c), (b) simulation.

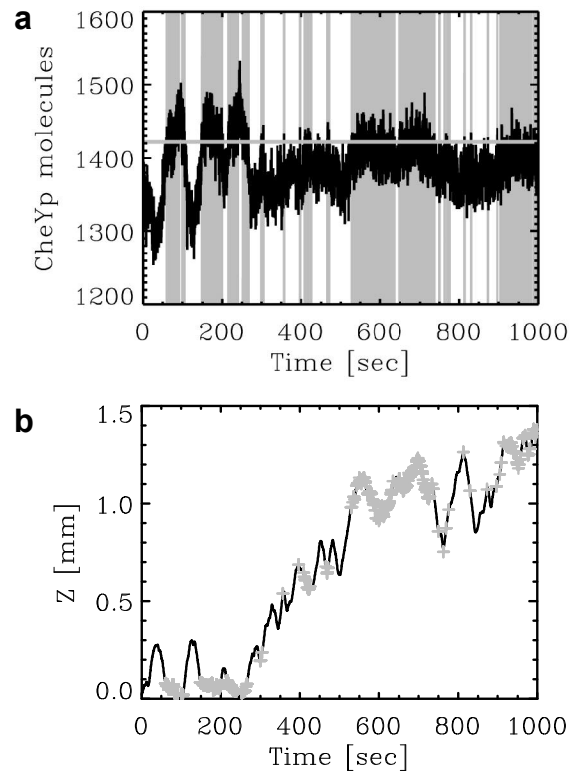


Fig. 3. Chemotactic response of a single synthetic cell to a constant gradient of aspartate $10^{-8} \nabla z$ M/ μ m. (a) Intracellular number of CheY-P molecules as function of time (black). The time series was smoothed with a sliding average of width 0.3 s. The motor switches when the number of CheY-P molecules crosses the threshold 1422 (horizontal grey line). [See also Section 3.3 and Table 1]. The vertical grey stripes indicate tumbling events. (b) Projection of the associated trajectory on the z axis. The grey crosses indicate tumbles.

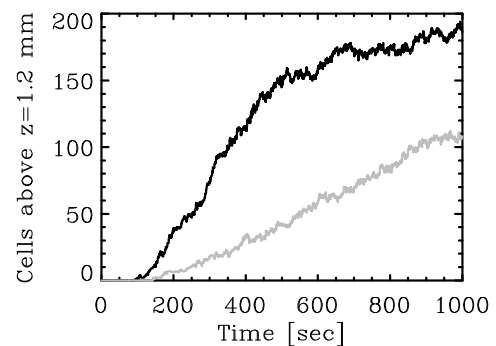


Fig.4. Average of the chemotactic response of 540 cells. Number of cells present in the half space $z > 1.2$ mm as function of time: (black) in the presence of a constant linear gradient of aspartate $10^{-8} \nabla z$ M/ μ m; (grey) in a medium free of aspartate. The cells are moving within a half infinite domain bounded by a reflecting boundary at $z=0$. The initial position of the cells is $x_0=y_0=0\mu$ m and $z_0=20\mu$ m.