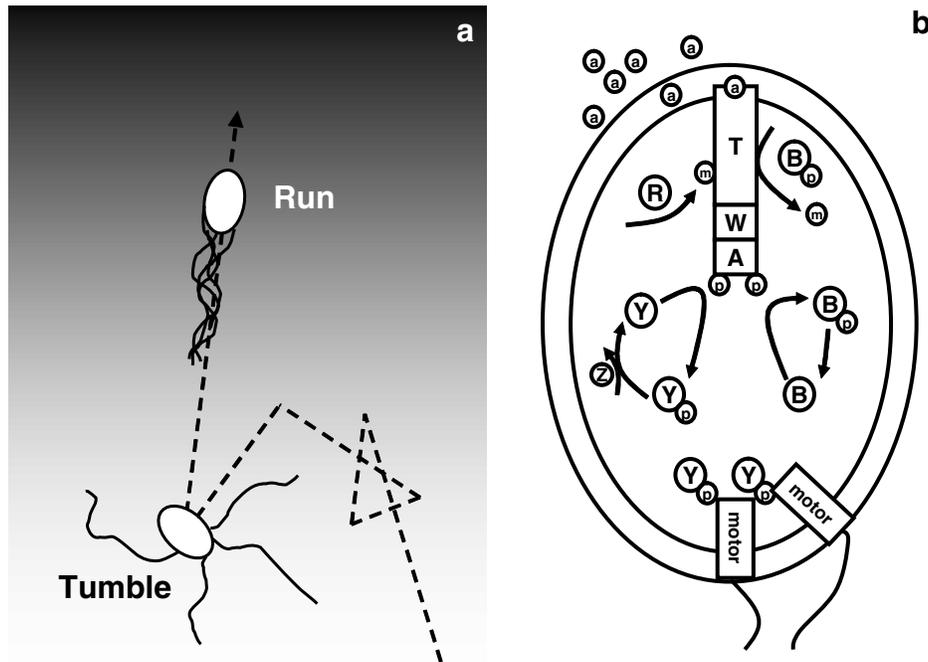


# Supplementary Information

## 1. The chemotaxis system



**Supplementary Fig. 1:** (a) Chemotactic swimming cell exposed to a linear gradient of chemo-attractant (shaded background). When the motors rotate CCW (see main text) the flagella form a corkscrew bundle and the cell runs. When the motors rotate CW, the flagella bundle flies apart and the cell tumbles erratically. (b) Schematic diagram of the chemotactic pathway. Receptor complexes consist of three Tar receptor dimers (T), two CheW (W), and two CheA (A). The activity of aspartate (a) receptor complexes is modulated by methylation (m) and demethylation respectively catalyzed by CheR (R) and CheB (B). The autophosphorylation of CheA (A), a histidine kinase, is controlled by the activity of the receptors. CheA catalyzes the phosphorylation of the response regulator CheY (Y). When CheY is phosphorylated (Y<sub>p</sub>), it binds to the cytoplasmic domain of the bacterial motor and induces tumbles. Dephosphorylation of CheY-P (Y<sub>p</sub>) is catalyzed by the phosphatase CheZ (Z).

## 2. Time stepping algorithm

In this version of AgentCell, agents are individual bacteria. A bacterium (*ChemotacticCell*) is equipped with a *step* method. The *step* algorithm is the following:

- (1) Ask the scheduler for the current time and calculate  $\Delta t$ , the time elapsed since the last call.
- (2) Using  $\Delta t$  as argument, call the step method of each object controlled by the cell. These methods advance the internal state of their object by the required amount  $\Delta t$ :

*chemotaxisNetwork.step*( $\Delta t$ )

*MotionStepper.step*( $\Delta t$ )

*motor.step*( $\Delta t$ )

*flagella.step*( $\Delta t$ )

- (3) Using the new cellular state, calculate the next  $\Delta t$  (i.e. when to schedule the next step)
- (4) Schedule the next call to itself at time  $t_{\text{current}} + \Delta t$ .

### 3. Time integration of the cell position and orientation

Let  $\mathbf{x}$  be a vector representing the position of the cell in the world (laboratory) coordinates and  $\mathbf{O}$  a 3x3 unitary matrix representing the orientation of the cell. Any point fixed in the cell body can be converted to the world (laboratory) coordinates using  $\mathbf{y} = \mathbf{O} \mathbf{y}_{\text{cell}} + \mathbf{x}$ , where  $\mathbf{y}_{\text{cell}}$  is the position of the point in body coordinates. The direction of motion  $\mathbf{n}$  is aligned with the z-axis of the cell coordinate system, i.e. it corresponds to the third column of the matrix  $\mathbf{O}$ . Neglecting inertia and translational Brownian motion, we can write the equations of motion as follows:

$$\begin{aligned} d\mathbf{x} &= v \mathbf{n} dt, \\ d\mathbf{O} &= \mathbf{O} \text{skew}\left(\sqrt{2D_r} d\mathbf{W}\right), \end{aligned}$$

where  $v$  is the constant speed during a run,  $D_r$  is the rotational diffusion coefficient,  $\mathbf{W}$  is a 3-dimensional Wiener process, and the skewed matrix is defined by:

$$\text{skew}(\mathbf{v}) = \begin{pmatrix} 0 & v_3 & -v_2 \\ -v_3 & 0 & v_1 \\ v_2 & -v_1 & 0 \end{pmatrix}.$$

For the numerical integration of the position we use the simple first order approximation:

$$\mathbf{x}(t + \Delta t) \cong \mathbf{x}(t) + v \mathbf{n} \Delta t.$$

The numerical integration for the time evolution of the orientation matrix  $\mathbf{O}$  is described in the supplementary information. To preserve the orthogonality of the orientation matrix  $\mathbf{O}$  we do not directly apply a numerical integration method to  $d\mathbf{O}$ . Instead we multiply  $\mathbf{O}$  by orthogonal matrices of infinitesimal rotations. Following Zou and Skel (2003), we define the angular vector increment

$$\Delta\boldsymbol{\varphi} = (\Delta\varphi_1, \Delta\varphi_2, \Delta\varphi_3)^T = \sqrt{2\Delta t D_r} \mathbf{N}$$

where

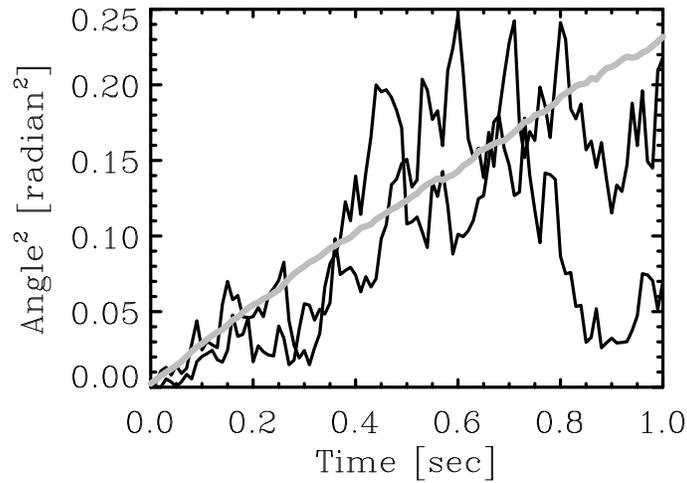
$$\sqrt{\Delta t} \mathbf{N} = \sqrt{\Delta t} (N_1, N_2, N_3)^T = \mathbf{W}(t + \Delta t) - \mathbf{W}(t)$$

is the increment of the 3D Wiener process and  $N_1, N_2$  and  $N_3$  are independent normal random variables. To integrate  $d\mathbf{O}$  we use:

$$\begin{aligned} \mathbf{O}(t + \Delta t) &= \mathbf{O}(t) \mathbf{R}_{\Delta\boldsymbol{\varphi}} \\ &\cong \mathbf{O}(t) \mathbf{R}_1(\Delta\varphi_1) \mathbf{R}_2(\Delta\varphi_2) \mathbf{R}_3(\Delta\varphi_3) \end{aligned}$$

with  $\mathbf{R}_{\Delta\boldsymbol{\varphi}}$  a rotation of angular vector  $\Delta\boldsymbol{\varphi}$  and  $\mathbf{R}_i(\Delta\varphi_i)$  a rotation of angle  $\Delta\varphi_i$  around the local  $i$ -axis. At each time step, the order in which the rotations  $\mathbf{R}_1, \mathbf{R}_2$ , and  $\mathbf{R}_3$  are applied is randomized.

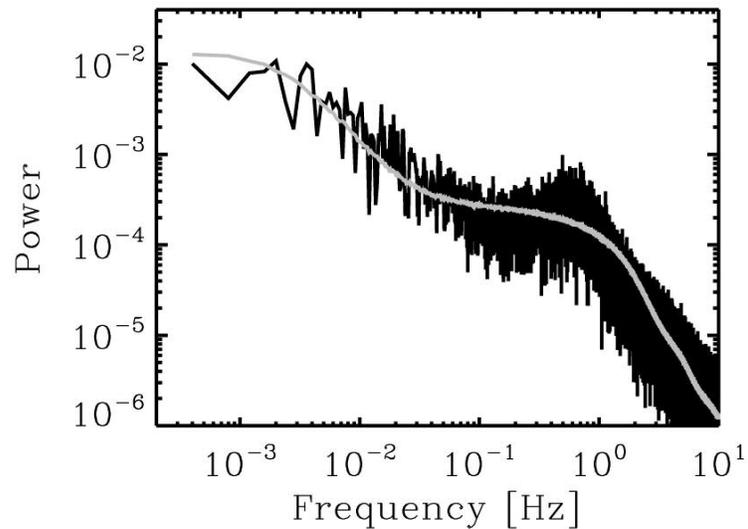
To test the time integration of the orientation, we simulated the evolution of 500 running cells over a period of 1 second. We used a constant time step of 0.01 second. The results are summarized in the Supplementary Figure 2, where we plot the squared angular deviation as function of time. The angular deviation is the angle between the initial and current directions of the cell:  $\theta = \text{acos}[\mathbf{n}(t) \cdot \mathbf{n}(t_0)]$ . As expected, the time evolution of the ensemble average is  $\langle \theta^2 \rangle = 4 D_r t$  (see e.g. Berg 1993).



**Supplementary Fig. 2.** Rotational diffusion: squared angular deviation as function of time for three independent cells (**black**). The **grey** line is the ensemble average over a population of 500 cells. As expected, the slope is  $4 D_r$  with  $D_r$ , the rotational diffusion coefficient. We used  $D_r = 0.062$  radians<sup>2</sup>/sec, the rotational diffusion coefficient for a sphere of radius  $1\mu\text{m}$  in a medium with viscosity  $\eta=0.027$  g(cm sec)<sup>-1</sup> and temperature  $T=305$  °K (see Table 1 in the main text and Berg 1993).

#### 4. Power spectrum of the flagella states of a single cell

Supplementary Figure 3 shows that our model reproduces the main spectral features of the laboratory data.



**Supplementary Fig. 3.** Power spectrum of the output of the chemotaxis system measured from one non-stimulated wild-type cell in a medium without attractant (Korobkova et al. 2004) (**black**). The output of the chemotaxis system is the binary time series of switching events recorded from one flagellar motor. The grey line represents the average of 1166 power spectra from the time series of the motor state of each individual cell from simulation 1.

## 5. Population behavior

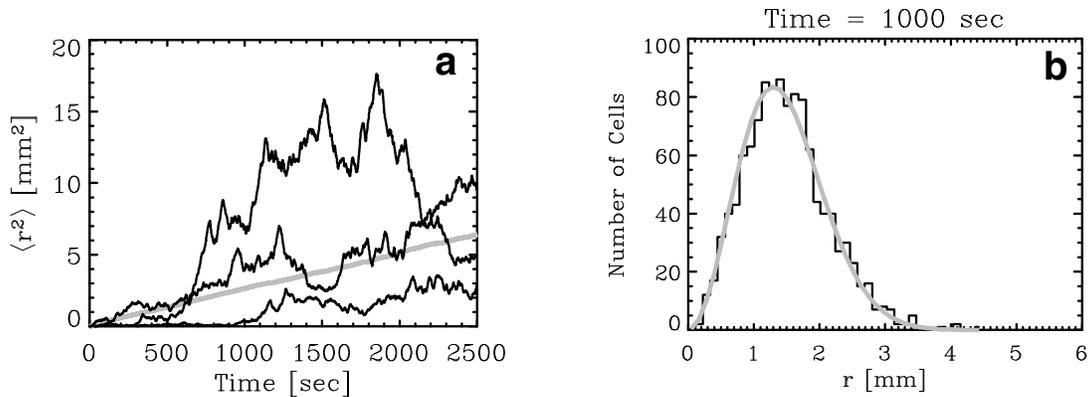
The square radial distance to the origin is plotted in Supplementary Figure 4 (panel **a**) for 3 individual cells (black curves) together with the population average (grey curve). The slope of the grey curve gives  $D = 4.21 \times 10^{-6} \text{ cm}^2/\text{s}$ . In this numerical setup (infinite domain, all cells started at the same point) the solution of the diffusion equation is simply

$$C(r,t) = \frac{N_0}{(4\pi D t)^{3/2}} \exp\left(-\frac{r^2}{4Dt}\right)$$

Where  $C$  is the cell concentration,  $r$  is the radial distance from the origin,  $t$  is the time and  $N_0$  is the total number of cells. The number of cells in a spherical shell comprised between the radii  $r$  and  $r+\Delta r$  is therefore

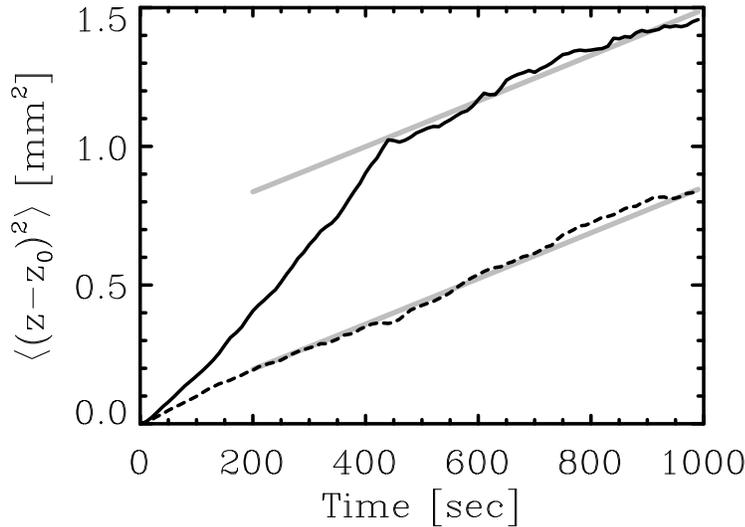
$$\Delta N(r,t) = 2[rC(r,t) - (r+\Delta r)C(r,t)] + N_0 \left[ \operatorname{erf}\left(\frac{r+\Delta r}{\sqrt{4Dt}}\right) - \operatorname{erf}\left(\frac{r}{\sqrt{4Dt}}\right) \right]$$

$\Delta N(r,t)$  is plotted as function of  $r$  for  $t=1000$  sec. together with the results from the simulation in Supplementary Figure 4 (panel **b**).



**Supplementary Fig. 4.** Population behavior: 3D diffusion of a population of 1166 cells in a medium without attractant. **(a)** Square distance from the origin as function of time for 3 independent cells. The grey line is the ensemble average over the population. The slope corresponds to a random motility coefficient  $D = 4.21 \times 10^{-6} \text{ cm}^2/\text{s}$ . **(b)** Number of cells in a spherical shell comprised between the radii  $r$  and  $r+\Delta r$  as function of radius  $r$  at time  $t=1000$  s. The bin size is  $\Delta r = 0.112$  mm. The grey curve corresponds to the analytical solution for  $\Delta N(r,t)$  (see text above).

As complementary illustration to the chemotaxis response of a digital population, we plot the population average of the square vertical distance,  $\langle (z-z_0)^2 \rangle$  in Supplementary Figure 5 (black full). After about 7 minutes, the receptors saturate and the slope changes to  $2 D t$  (grey lines), corresponding to the diffusion rate in the absence of aspartate. For comparison, the result obtained for a medium without aspartate is also plotted (Fig. 5, dashed line).



**Supplementary Fig. 5.** Response of a population of 540 cells to aspartate. Plot of the square vertical distance as function of time: (**black full**) in the presence of a constant linear gradient of aspartate  $10^{-8} \nabla z$  M/ $\mu\text{m}$ ; (**black dashed**) in a medium free of aspartate. The slope of the grey lines is  $2D$ .  $D$  is the random motility coefficient measured in Supplementary Fig. 4. The cells are swimming within a half infinite domain bounded by a reflecting boundary at  $z=0$ . The initial position of the cells is  $x_0=y_0=0\mu\text{m}$  and  $z_0=20\mu\text{m}$ .

## 6. Scalability of AgentCell on the Jazz cluster at Argonne National Laboratory

Because the current version of AgentCell does not include cell-to-cell communications, each single cell run is executed as an independent process: cells run on separate processors and dump data in separate files. Thus, the model has theoretically linear speedup for execution of cells in parallel. In practice, there are two potential causes of resource contention during large-scale cluster execution. One is the file system and the other is the associated network usage. We used the Argonne Jazz cluster's Ethernet network (Metcalfe and Boggs 1976, IEEE 2002) to initialize the runs and to retrieve the data files for each individual cell at the end of execution. The retrieval was done using the Parallel Virtual File System (PVFS 2005, Haddad 2000). Both Ethernet and PVFS are highly scalable in practice (Francis et al. 1989, Garg and Mache 2001).

## 7. How to use AgentCell

The following is the content of the README.txt file included with the AgentCell distribution. It explains how to unpack, install AgentCell and how to run an example simulation.

USAGE ON A LINUX BOX:

A Unpack:

=====

- 1) unpack everything in a directory of your choice.
- 2) `cd agentcell`
- 3) `chmod +x run* javacompile cleanoutput`
- 4) if the directories `run` and `OutputCluster` don't exist, create them:  
`mkdir run`

mkdir OutputCluster

B Compile the stochsim library.

=====

This will create the file libstochsim.so

- 1) cd lib/stochsim
- 2) make clean
- 3) make
- 4) make clean
- 5) cd ../..

C Compile the java sources.

=====

- 1) ./javacompile

D Run one cell within a linear gradient of attractant for 5 seconds

=====

- 1) ./runbatch 1
- 2) if you have 2 CPUs you can start a 2nd independent cell:  
./runbatch 2
- 3) if you have 3 CPUs you can start a 3rd independent cell:  
./runbatch 3
- ...

To submit several cells simultaneously on a cluster (distributed environment that runs the bps queue system), use the runpbs script.

To submit several cells on a multi-CPU machine with shared memory, use the runnopbs script.

E To create your own run

=====

- 1) cd edu/uchicago/agentcell/runs
- 2) cp LinearGradientAveragedMotorBergTumbleSwimmingFlagella.java myrun.java
- 3) edit myrun.java (modify boundaries, cell components, ...)  
replace LinearGradientAveragedMotorBergTumbleSwimmingFlagella with Myrun
- 4) go back to the root directory (cd ../../../../..)
- 5) ./javacompile
- 6) edit the file runbatch and in it replace  
LinearGradientAveragedMotorBergTumbleSwimmingFlagella  
with  
Myrun
- 7) remove previous run outputs:  
rm -f run/\*  
rm -f OutputCluster/\*
- 8) ./runbatch 1

## 8. Supplementary References

- Francis, S., Frost, V., and Soldan, D. (1989) Measured Ethernet Performance for Multiple Large File Transfers, in *Proceedings of the 14th Conference on Local Computer Networks 1989 IEEE*, pp. 323-327
- Garg, S., and Mache, J. (2001) Performance Evaluation of Parallel File Systems for PC Clusters and ASCI Red, in *3rd IEEE International Conference on Cluster Computing, 2001 IEEE*, pp. 172-177
- Haddad, I. F. (2000), PVFS: A Parallel Virtual File System for Linux Clusters, *Linux Journal*, **80**, <http://www.linuxjournal.com/article/4354>
- IEEE, (2002) IEEE Standard 802.3-2002: Carrier Sense Multiple Access with Collision Detection (CSMA/CD), Access Method and Physical Layer Specifications, 1562.
- Metcalfe, R. and Boggs, D. (1976) Ethernet: distributed packet switching for local computer networks. *Communications of the ACM*, **19**, 395-404.
- Parallel Virtual File System Team Home Page (2005), <http://www.pvfs.org/>
- Zou Gang, Skeel Robert D. (2003) Robust Biased Brownian Dynamics for Rate Constant Calculation, *Biophys. J.*, **85**, 2147-2157