Journal of Microbiological Methods 175 (2020) 105918

Contents lists available at ScienceDirect

Journal of Microbiological Methods

journal homepage: www.elsevier.com/locate/jmicmeth

TaxisPy: A Python-based software for the quantitative analysis of bacterial chemotaxis

Miguel Á. Valderrama-Gómez[a,*,1], Rebecca A. Schomer[a,1], Michael A. Savageau[a,b], Rebecca E. Parales[a]

[a] Department of Microbiology & Molecular Genetics, College of Biological Sciences, University of California, Davis, USA
[b] Department of Biomedical Engineering, University of California, Davis, USA

ARTICLE INFO

Keywords:
Cell tracking software
Quantitative chemotaxis
Video analysis software

ABSTRACT

Several species of bacteria are able to modify their swimming behavior in response to chemical attractants or repellents. Methods for the quantitative analysis of bacterial chemotaxis such as quantitative capillary assays are tedious and time-consuming. Computer-based video analysis of swimming bacteria represents a valuable method to directly assess their chemotactic response. Even though multiple studies have used this approach to elucidate various aspects of bacterial chemotaxis, to date, no computer software for such analyses is freely available. Here, we introduce TaxisPy, a Python-based software for the quantitative analysis of bacterial chemotaxis. The software comes with an intuitive graphical user interface and can be accessed easily through Docker on any operating system. Using a video of freely swimming cells as input, TaxisPy estimates the culture's average tumbling frequency over time. We demonstrate the utility of the software by assessing the effect of different concentrations of the attractant shikimate on the swimming behavior of *Pseudomonas putida* F1 and by capturing the adaptation process that *Escherichia coli* undergoes after being exposed to L-aspartate.

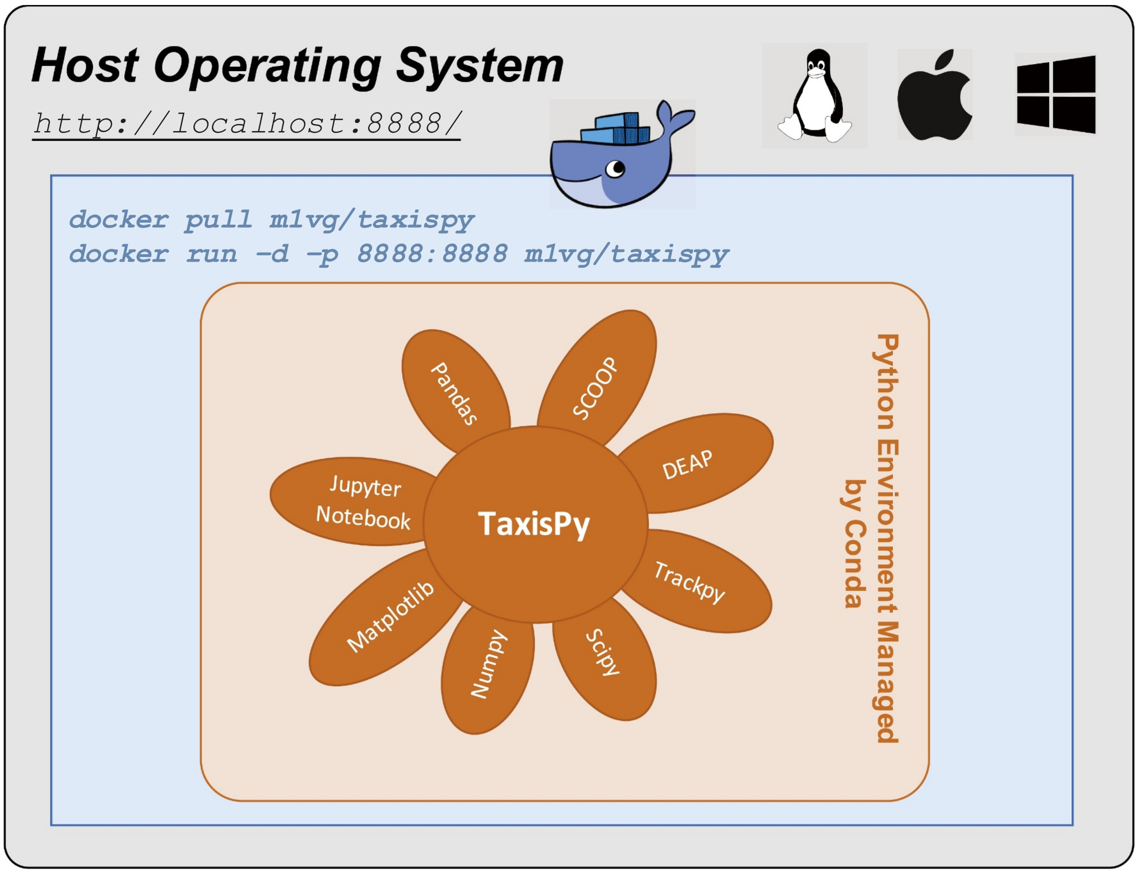
1. Introduction

Environmental conditions such as temperature, light intensity and chemical composition influence the ability of an organism to grow, reproduce and survive. Many bacteria have evolved intricate mechanisms to sense these and other external stimuli. These signals are processed by the cell and directly affect the functioning of its motility apparatus, thus leading to an informed displacement towards a beneficial environment or away from unfavorable conditions. Chemotaxis refers to a modification in the motility pattern of an organism in response to a change in the chemical composition of its environment. Motile bacteria utilize chemotaxis for a variety of purposes. For example, bacteria move towards chemicals that serve as carbon and energy sources, nitrogen sources, and electron acceptors (Matilla and Krell, 2017; Alexandre, 2010). Similarly, pathogenic and mutualistic bacteria that associate with specific plants and animals use chemotaxis to sense chemical signals released by their hosts (Matilla and Krell, 2018; Scharf et al., 2016). In addition, there is evidence that chemotactic bacteria that are capable of degrading toxic xenobiotic pollutants are more efficient at biodegradation than nonmotile or nonchemotactic bacteria (Parales et al., 2015). Flagellated bacteria such as *Escherichia coli* and *Pseudomonas putida* swim in liquid environments by rotating their helical flagella. A change in the swimming direction is achieved by the intermittent change of the direction of flagellar rotation, which leads to tumbles that usually last only a fraction of a second (Webre et al., 2003). The swimming pattern of a cell can be quantitatively characterized by its tumbling frequency (Alon et al., 1999; Pohl et al., 2017). In the absence of chemical attractants, the swimming pattern of motile bacteria is described as a "random walk", which is characterized by short periods of smooth swimming (often termed "runs") and a high tumbling frequency, with typical values of 0.44 changes of direction per second for *E. coli* (Alon et al., 1999; Staropoli and Alon, 2000). After a chemical attractant is sensed, chemotactic bacteria decrease their tumbling frequency to values lower than 0.05 per second (Alon et al., 1999). As a result, cells in a culture effectively swim towards higher concentrations of the attractant following straight paths.

Over the last decades, qualitative and quantitative methods for the study of bacterial chemotaxis have been developed. Some representative protocols include capillary assays (Pfeffer, 1884; Adler, 1969; Adler, 1973), chemical-in-plug assays (Tso and Adler, 1974) and

* Corresponding author.
E-mail address: mavalderramagomez@ucdavis.edu (M.Á. Valderrama-Gómez).
[1] Authors equally contributed to this work

https://doi.org/10.1016/j.mimet.2020.105918
Received 7 February 2020; Received in revised form 9 April 2020
Available online 06 June 2020



```
docker pull m1vg/taxispy
docker run -d -p 8888:8888 m1vg/taxispy
```

Fig. 1. Software architecture. TaxisPy bundles several Python packages that together allow a computer-assisted, quantitative analysis of bacterial chemotaxis. TaxisPy's user interface is based on Jupyter widgets and can be accessed from Jupyter Notebooks on any web browser, e.g., Google Chrome. TaxisPy's ability to identify cells and generate trajectories is enabled by the Python package Trackpy (Allan et al., 2019). Pandas, Numpy and Scipy are additional packages that provide TaxisPy with computational objects and functions required to store, filter and process the data generated by Trackpy. The Python packages DEAP (Fortin et al., 2012) and Scoop (Hold-Geoffroy et al., 2014) are used to estimate parameter values for the identification of tumbles. All plotting routines are supported by Matplotlib.

computer-based video analyses of swimming cells (Berg and Brown, 1972). See Ditty and Parales (2015) for an overview of these methods. While qualitative methods are fast and usually do not involve major technical difficulties, quantitative methods such as the quantitative capillary assay are tedious and time-consuming. Computer-based video analysis of swimming bacteria represents a valuable method to directly and quantitatively assess the chemotactic response of bacteria. Even though multiple studies have used this approach to elucidate various aspects of bacterial chemotaxis (Harwood et al., 1989; Alon et al., 1998; Alon et al., 1999; Staropoli and Alon, 2000), to date, no computer software for such analyses is freely available. Here, we introduce TaxisPy, a Python-based software for the quantitative analysis of bacterial chemotaxis. The software comes with an intuitive graphical user interface that is especially suited for users with limited programming knowledge. It systematically addresses typical difficulties associated with the customized use of cell tracking software such as filtering of atypical cell trajectories and determination of parameter values for the identification of tumbles. TaxisPy can be easily accessed through Docker on any operating system.

2. Materials and methods

2.1. Bacterial strains, media composition and attractants

The responses of *E. coli* K12 strain RP437 [F$^-$ *thr-1 leuB6 his-4 metF159 thi-1 ara-14 lacY1 mtl-1 xyl-5 rpsL136 tonA31 tsx-78 eda-50*] (Parkinson, 1978) and *Pseudomonas putida* F1 (Finette et al., 1984; Gibson et al., 1970) –both motile organisms used as models for studying chemotactic behavior– to chemical attractants were examined in this study. All cells were grown in Luria-Bertani medium (LB; Sambrook et al., 1989) or minimal medium (MSB; Stanier et al., 1966) at 30 °C in a shaking incubator.

2.2. Behavioral assays

Pseudomonas putida cultures were grown for approximately 18 h in minimal MSB medium containing 10 mM succinate at 30 °C. One hundred microliters of the culture were sub-cultured into 15 mL minimal MSB containing 10 mM succinate and 5 mM shikimate for induction of chemotactic behavior towards aromatic acids (Luu et al., 2015). After approximately 6 h of growth, the cells were harvested at mid-exponential phase ($OD_{600} = 0.4–0.6$) by centrifugation at 5000 rpm for 10 min at room temperature. The cells were gently washed in 15 mL of chemotaxis buffer (CB; 50 mM potassium phosphate buffer pH 7.0, 10 μM disodium EDTA, 0.05% glycerol) and re-centrifuged for 10 min at 5000 rpm (Parales et al., 2000). Finally, the cells were resuspended in 15 mL of aerated CB. Cultures were diluted 2- to 3-fold with CB to limit overlapping trajectories (Supplemental Information, Section 7.1). For analysis of the swimming pattern of *P. putida*, 10 μL of the cell suspension was placed on a glass slide (Fisher Scientific) and mixed with 1 μL of various concentrations of shikimate in CB to attain final concentrations of 0 μM, 10 μM, 50 μM, 100 μM, and 1 mM. Two seconds after the addition of the attractant, video recording was initiated. Recording of the samples was performed with an Infinity Lite microscope camera (Lumenera, Ottawa, ON, Canada) mounted to a Nikon eclipse TE2000-S inverted microscope, using a magnification of 400×. The swimming behavior was recorded for 1 min at a speed of 20.3 frames per second using the software Infinity Capture 6.5.4. These videos were then analyzed with TaxisPy as described below to estimate the tumbling frequency of the cell populations.

Samples for analyzing the adaptation of *E. coli* after stimulation with L-aspartate were generated using a different set-up. Cells were grown overnight in LB medium before being harvested by centrifugation (10,000 rpm, 1 min) and washed with an equal volume of MSB medium. Cells were resuspended in an equal volume of MSB medium. Two hundred microliters of washed culture were inoculated into 15 mL of minimal MSB medium containing 12 mM glucose. When the cultures reached mid-exponential phase (OD_{600} = 0.4–0.6), 1 mL aliquots of culture were harvested by centrifugation at 5000 rpm for 10 min at room temperature and gently resuspended in 1 mL of CB. At regular time intervals, 10 μL samples of *E. coli* were removed and observed with a Nikon eclipse TE200-S inverted microscope at 400× magnification. Using the Infinity Lite mounted camera, 10 s videos were recorded at a speed of 20.3 frames per second. The chemotactic response was initiated by adding L-aspartate to the resuspended cultures at final concentration of 1 mM. To keep cells motile during the experiment, cells were incubated at 30 °C in a shaking incubator (200 rpm). These videos were then analyzed with TaxisPy as described below to estimate the tumbling frequencies of the cell populations.

2.3. Software and general workflow

TaxisPy integrates several Python packages to enable the estimation of the average tumbling frequency of bacteria in culture (Fig. 1). Its intuitive graphical user interface is based on Jupyter widgets and contains six different tabs (Fig. 2). In order to streamline its distribution, especially among users with limited programming knowledge, TaxisPy can be accessed via a Docker image. Docker is a computer program that performs operating-system-level virtualization and is used to run software packages called containers. Containers bundle their own application, tools, libraries and configuration files and are created from images. Docker has been increasingly used to distribute software across different operating systems, partly because it guarantees the reproducibility of computational workflows performed within its containers (Beaulieu-Jones and Greene, 2017). This feature renders Docker an appealing tool (Boettiger, 2015) to distribute a variety of research software. Some recent examples include stochastic simulators (Drawert et al., 2016), generators of bioinformatic workflows (Hung et al., 2019) and tools for the mechanistic analysis of biochemical networks (Valderrama-Gómez et al., 2020) among many others.

To download TaxisPy via Docker, four steps are required:

1. Install Docker on your computer. Docker is supported by all major operating systems: Linux, MacOs and Windows. Refer to the installation instructions in the Supplemental Information (Section 3) for further details.
2. Download the latest TaxisPy image by typing the following command in a terminal window (Mac, Linux) or command prompt window (Microsoft Windows):

```
docker pull m1vg/taxispy
```

3. Start a Docker container to access TaxisPy by typing the following command in the same window:

```
docker run -d -p 8888:8888 m1vg/taxispy
```

This command will create a container without the ability to read or write files on the host computer (i.e., your computer). Files created within the container will be lost after the container is stopped. In order

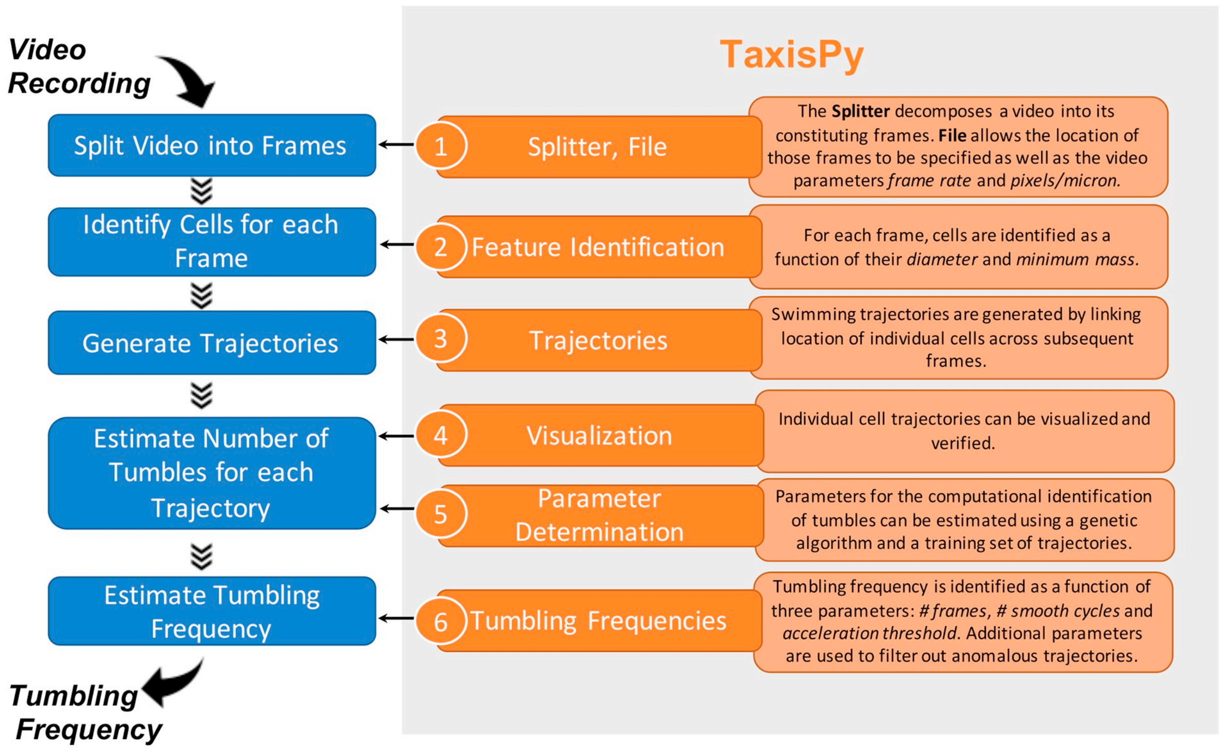

Video Recording
TaxisPy
Split Video into Frames
Identify Cells for each Frame
Generate Trajectories
Estimate Number of Tumbles for each Trajectory
Estimate Tumbling Frequency
Tumbling Frequency
1 Splitter, File
2 Feature Identification
3 Trajectories
4 Visualization
5 Parameter Determination
6 Tumbling Frequencies
The Splitter decomposes a video into its constituting frames. File allows the location of those frames to be specified as well as the video parameters frame rate and pixels/micron.
For each frame, cells are identified as a function of their diameter and minimum mass.
Swimming trajectories are generated by linking location of individual cells across subsequent frames.
Individual cell trajectories can be visualized and verified.
Parameters for the computational identification of tumbles can be estimated using a genetic algorithm and a training set of trajectories.
Tumbling frequency is identified as a function of three parameters: # frames, # smooth cycles and acceleration threshold. Additional parameters are used to filter out anomalous trajectories.

Fig. 2. From a video recording to the bacterial tumbling frequency using TaxisPy. Five different tasks –represented by the blue rectangles– are involved in estimating the tumbling frequency of bacteria in culture from a video of their swimming behavior. TaxisPy's user interface consists of six sequentially arranged tabs –represented by the orange rectangles– that provide required functionalities to perform each one of these tasks. Even though several parameters are involved in various steps of the analysis, TaxisPy provides necessary tools to identify appropriate values for those parameters. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

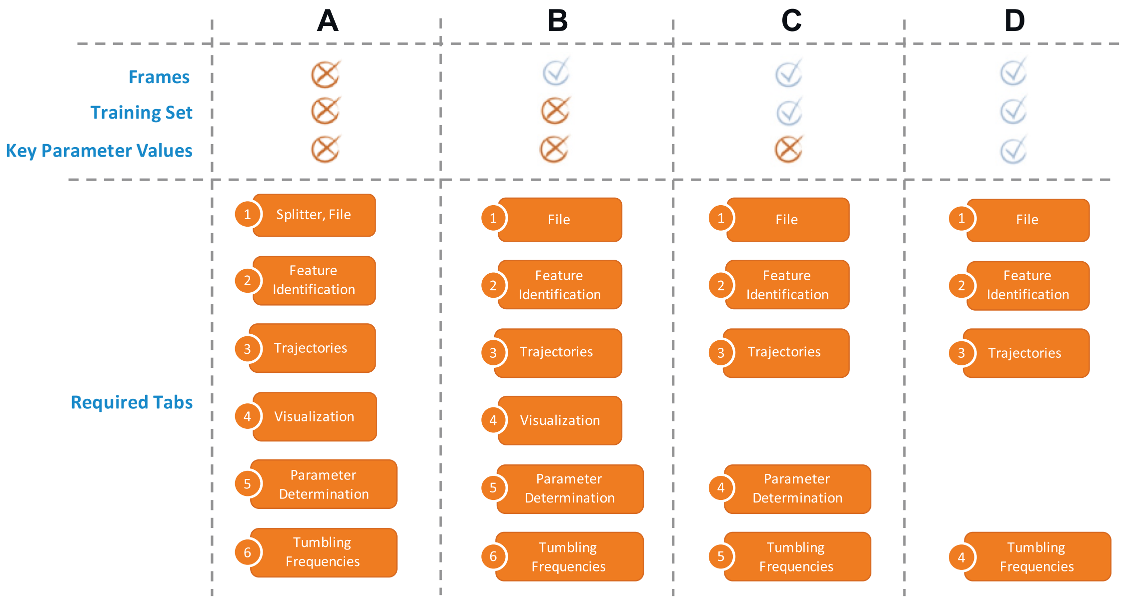


Fig. 3. Workflows supported by TaxisPy. Different workflows can be followed when working with TaxisPy. The most appropriate workflow will depend on the availability of a set of frames, a user-provided training set, and values of key parameters for the identification of tumbling frequencies. A) If none of the latter is available, the standard workflow (see also Fig. 2) should be followed. This workflow starts by splitting a given video into its constituting frames and ends by estimating the average tumbling frequency. Refer to the tutorial contained in the TaxisPy Docker image under /Tutorial for a step by step description of the general workflow. B) If a set of frames is already available, the first step of the standard workflow can be skipped, i.e., using the splitter. C) If a set of frames, along with a training set are available, key parameter values for the determination of tumbling frequencies can be estimated after the frames have been loaded into TaxisPy and cellular trajectories have been identified by the software. The training set can be either manually entered or conveniently loaded from an Excel (.xlsx) file –the file should contain two columns, one with the header "Trajectory" and one with the header "Tumbles". D) If frames, along with a training set and the resulting values for key parameters are available, the average cellular tumbling frequency can be directly estimated by TaxisPy once the frames have been loaded into TaxisPy and cellular trajectories have been identified by the software. This workflow involves the utilization of the tab *File* to load the frames into TaxisPy, the tabs *Feature Identification* and *Trajectories* to automatically identify cellular trajectories, and the tab *Tumbling Frequencies* to estimate the average tumbling frequency.

to grant access to files on the host computer, the previous command should be replaced by the following command:

```
docker run -d -p 8888:8888 –mount type=bind,source=/
Users,target=/Documents/host m1vg/taxispy
```

Note that the lines of code above/below correspond to one single command. Windows users should use:

```
docker run -d -p 8888:8888 –mount type=bind,source=//
c/Users,target=/Documents/host m1vg/taxispy
```

4. Access TaxisPy by entering the following address in any internet browser (e.g., Google Chrome):

   http://localhost:8888/

Refer to additional instructions to start Docker containers as well as frequently asked questions in the Supplemental Information (Sections 3 to 6).

Quantitatively characterizing bacterial swimming behavior of a given culture by means of its tumbling frequency involves five consecutive steps, as shown in Fig. 2. First, a video registering freely swimming cells is split into its constituting frames. Each frame represents a snapshot of the culture at a given time point. Splitting can be performed using the software FFmpeg (it supports a wide range of video formats), for which a simple user interface is provided within the TaxisPy Docker image; or by any other software designed for that purpose. Then, cells present in each frame are located and swimming trajectories identified by linking the position of individual cells through subsequent frames. Within TaxisPy, this functionality is provided by Trackpy 0.4.2 (Allan et al., 2019), a Python package that implements feature-finding and linking algorithms originally introduced by Crocker and Grier (1996). The next step consists of calculating the number of tumbles for each trajectory. Previous reports have used motion properties of individual trajectories – e.g., acceleration and velocities– to calculate the number of tumbles for each cellular trajectory. The underlying idea is that a cell decreases its linear velocity and increases its absolute acceleration below/above certain thresholds while tumbling. Specific threshold values were empirically determined by the authors of those studies (Berg and Brown, 1972; Sager et al., 1988; Harwood et al., 1989; Amsler, 1996; Alon et al., 1998). It is clear that these values are a function of a variety of biological (i.e., bacterial strain, growth conditions, optical density, etc.) and technical (i.e., frame rate, signal smoothing) factors, which restricts the utility of such thresholds to the specific conditions for which they were estimated. To calculate the number of tumbles for a given cellular trajectory, TaxisPy requires the values for three key parameters: a threshold value for the absolute acceleration and values for two parameters controlling the smoothing of cellular trajectories: *# Frames* and *# Smooth* (see Table S1 in the Supplemental Information). Optimal values for these parameters are estimated based on a user-provided training set, so that the squared difference between the observed and the calculated number of tumbles is minimized. The condition-specific training set consists of a set of cellular trajectories with known changes of direction, and its generation is fully supported by various functionalities contained within TaxisPy. A genetic algorithm, provided by the Python package DEAP (Fortin et al., 2012), is used to solve the optimization problem required to estimate key parameter values. The final step in the characterization of the bacterial swimming behavior in a culture involves the determination of its average tumbling frequency. This value is calculated by dividing the total number of tumbles estimated for a set of trajectories over the total duration of those trajectories. Alternatively, the tumbling frequency can be calculated for each individual trajectory and an average over all trajectories can be used to estimate the average tumbling frequency of the culture. TaxisPy offers the flexibility to select the desired calculation method. In this work, reported tumbling frequencies were calculated using the first method, which divides the total number of tumbles over the total duration of all trajectories. In order to exclude anomalous trajectories that might bias the average tumbling frequency calculated for a set of trajectories, TaxisPy offers a series of filters to remove trajectories of nonmotile cells, trajectories exhibiting a too high number of tumbles (e.g., "stuck" cells that are trying to detach) or trajectories with a linear velocity that is abnormally low.

Depending on the availability of a set of frames, a condition-specific training set, and key parameter values for the identification of tumbling frequencies, various workflows are possible when working with TaxisPy (Fig. 3). There are two advantages associated with the ability to follow different workflows. First, the results of a given motion analysis can be quickly reproduced if a set of frames and values for key parameters are provided (Fig. 3D). Second, the steps required to estimate the average tumbling frequency from a video do not need to be executed in a single, uninterrupted session. This allows multiple users to contribute to the analysis of a given video (e.g., user 1 generates the training set, user 2 determines optimal parameter values, which are employed by user 3 to calculate the average tumbling frequency) and allows the possibility to resume the analysis at any stage (i.e., workflow B, C or D) if TaxisPy unexpectedly crashes.

2.4. Key parameter values required by TaxisPy

A comprehensive list of all parameters required by TaxisPy, along with a short description and nominal values is provided in Table S1 of the Supplemental Information. Eleven key parameters are involved in different stages of the analysis. These include parameters such as *Diameter* and *Min. Mass,* which are used in the Feature Identification tab and are directly passed to Trackpy (Allan et al., 2019) for the identification of individual cells in each frame. A nominal value of 25 pixels for *Diameter* and 2000 for *Min. Mass* were used for all video analyses of both *P. putida* and *E. coli* cultures. TaxisPy offers an intuitive way to identify optimal values for these two parameters by visually inspecting the ability of the software to correctly identify cells in three different frames, corresponding to the first, middle, and last frame of a given set. Cells identified in these frames are enclosed by a blue circle by TaxisPy. Appropriate values for *Diameter* and *Min. Mass* will maximize the number of cells correctly identified by TaxisPy. Refer to Section 7 in the Supplemental Information for suggestions on optimal video acquisition and image analysis.

A second group of key parameters consists of *# Trajectories*, *Trajectory #* and *# Chng. Dir*. They refer to the number of trajectories within a given user-defined training set, the ID of each trajectory and the observed number of changes of direction or tumbles for each trajectory, respectively. These parameters determine the training set that is passed to the genetic algorithm implemented by DEAP (Fortin et al., 2012). The solution of the optimization problem corresponds to optimal values for the two parameters controlling the smoothing routine (*#Frames* and *#Smooth)* and a value for the acceleration threshold (*Acc. Thrhld.*), which are used to identify the number of tumbles of all cellular trajectories. Briefly, a genetic algorithm is a heuristic search and optimization technique inspired by natural evolution. First proposed by John Holland (Holland, 1975), genetic algorithms have been successfully applied to a wide range of real-world problems of significant complexity (McCall, 2005). They use a highly abstract version of the evolutionary process to *evolve* solutions to a given optimization problem. It starts with a randomly generated population of artificial "chromosomes" and carries out a process of fitness-based selection, recombination and mutation to produce an offspring population. This process is iterated for a certain number of generations. In our specific context, an artificial chromosome corresponds to a numerical vector containing values for the key parameters [*# Frames*, *# Smooth*, *Acc.Thrhld*] and its fitness is defined by the sum of squared differences between the observed and the calculated number of tumbles for each trajectory of the training set. The genetic algorithm implemented by DEAP used a population size of 100 individuals (artificial chromosomes) and 5 generations. The initial population was generated at random, using ranges for *Acc. Thrhld* of 0 to 1000 $\mu m/s^2$ and *#Frames* and *#Smooth* of 1 to 5. To increase variability within the population, we applied a two-point crossover and gaussian mutation with mean of 0 and standard deviation of 0.2. The independent probability for each attribute of the population to be mutated was 0.2. This configuration of the genetic algorithm exhibited a good performance for the training sets used in this study but can be customized if necessary. Population size, number of generations and bounds of key parameters can be customized from the tab Parameter Determination. The training set used by DEAP can be easily generated by the user employing the tabs Trajectories and Visualization. Refer to Tables S2 to S6 in the Supplemental Information for training sets used to identify the acceleration threshold and smoothing parameters for the *P. putida* experiments and to Tables S8 to S10 for training sets used for the *E. coli* experiments. Optimal parameter values resulting from these training sets are summarized in Table S7 for the *P. putida* videos and in Table S11 for *E. coli*.

An additional group of key parameters is located within the tab Tumbling Frequencies. These parameters control the way TaxisPy filters out anomalous trajectories and calculates average tumbling frequencies. The parameters *Dsplcmt, %*, *Velocity* and *Max. Chng. Dir.* are used to sort out short trajectories, trajectories of cells swimming with too low linear velocities and trajectories exhibiting an excessive number of turns, respectively. Nominal values of 10%, 4 μm/s, and 10 tumbles were used to analyze all data presented in this study. Finally, the parameter *T. Int. (s)* is used to specify the time intervals used by TaxisPy to calculate the temporal evolution of the cellular tumbling frequency. A value of 5 s was used for the analysis of the *P. putida* experiments, while a value of 10 s was used for *E. coli*.

3. Results

In order to demonstrate the utility of TaxisPy, the chemotactic responses of two different microorganisms after stimulation with chemical attractants were video recorded and analyzed. Using the frames of the recorded videos as input, TaxisPy allowed the quantification of the effect of shikimate on the tumbling frequency of *P. putida* and the visualization of the adaptation process of *E. coli* after exposure to 1 mM L-aspartate. The following results were generated using parameter values listed in Table S7 for *P. putida* F1 and Table S11 for *E. coli*. Refer to ReadMe Notebooks contained in the Docker image under /Ecoli_Aspartate and /Pseudomonas_Shikimate to reproduce individual results.

3.1. Characterization of the chemotactic response of P. putida

P. putida F1 was exposed to four different concentrations of shikimate. Chemotaxis buffer without shikimate was used as the control condition. The response of the culture under each condition was video recorded in triplicate for sixty seconds and TaxisPy was subsequently used to calculate the average tumbling frequency under each condition as described in the Methods section. Fig. 4 summarizes our findings. As expected, the cellular tumbling frequency decreased from its basal value of 0.42 tumbles/s under unstimulated conditions to a low value of 0.023 tumbles/s for a shikimate concentration of 1 mM.

The *minimum* average cellular tumbling frequency exhibited by *stimulated* cultures within the first 20 s of each video was used in Fig. 4A to assess the effect of different concentrations of shikimate on the swimming pattern. This value was used instead of the *average* tumbling frequency over the same period of time because it better captured the reduction of the culture's tumbling frequency after exposure to the chemical attractant. Note that the only difference between the *minimum*

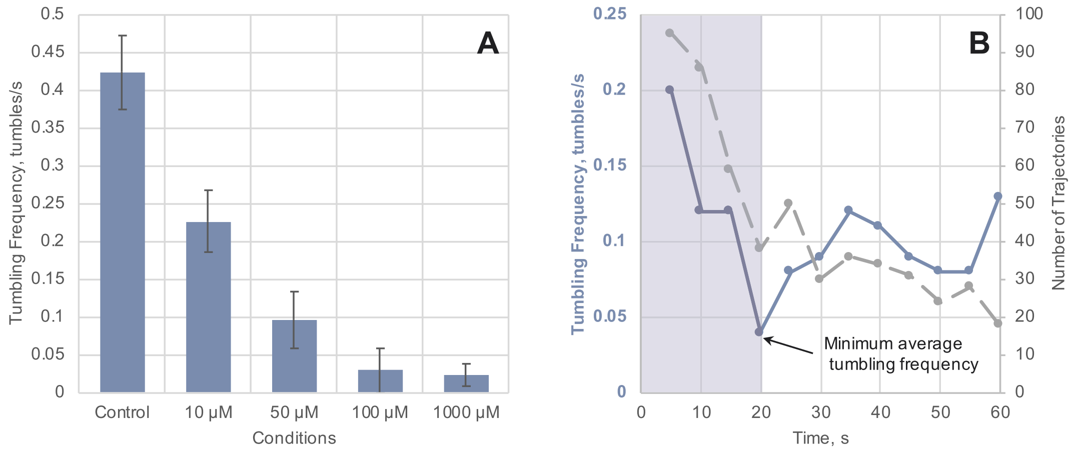


Fig. 4. Tumbling frequency of *P. putida* for different shikimate concentrations. A) Tumbling frequency exhibited by the cells as a function of the shikimate concentration. After stimulation with shikimate, the swimming pattern was characterized by the *minimum* average tumbling frequency exhibited by the cells during the first 20 s and calculated using time intervals of 5 s. For the control cultures (CB only), the swimming pattern was characterized by the average tumbling frequency calculated for the first 20 s of each video. Refer to main text for details. The number of trajectories used to calculate tumbling frequencies for each video ranged from 257 to 23. Error bars represent one standard deviation of the cellular tumbling frequency calculated from three different videos. B) Temporal evolution of the tumbling frequency of *P. putida* after exposure to 1 mM shikimate is represented by the blue solid line. Average frequencies were calculated using intervals of 5 s; the corresponding data point is placed at the end of each interval. The grey dashed line represents the number of cellular trajectories used by TaxisPy to calculate reported average frequencies. The shaded blue area represents the first 20 s of the video used to determine the *minimum* average tumbling frequency. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

average tumbling frequency and the *average* tumbling frequency is the time period and consequently the number of trajectories used for its calculation. While the average tumbling frequency considers all detected trajectories during the first 20 s, the *minimum* average tumbling frequency involves the calculation of four different average tumbling frequencies, which are calculated using trajectories detected in the time ranges $0 \leq t < 5$ s (f_1), $5 \leq t < 10$ s (f_2), $10 \leq t < 15$ s (f3) and $15 \leq t < 20$ s (f_4). The minimum average tumbling frequency corresponds to the minimum numerical value of all four frequencies: *min*(f_1, f_2, f_3, f_4). This procedure is justified in Fig. 4B, where the temporal evolution of the tumbling frequency after addition of 1 mM of shikimate is shown (see four blue data points within the blue shaded area). While the average tumbling frequency for the first 20 s corresponds to 0.13 tumbles/s, the minimum average tumbling frequency within the same period of time corresponds to 0.04 tumbles/s. Note that this tumbling frequency was calculated using trajectories detected in the period of time between 15 and 20 s. The origin of the discrepancy between these two values can be attributed to temporal processes involved in the diffusion of the chemical attractant in the culture's medium, as well as to temporal processes involved in cellular sensing and response to the chemical gradient. Note that the analysis was restricted to the initial 20 s of the video instead of its total duration (see blue rectangle in Fig. 4B). There are two reasons for this. The first one is related to the adaptation process that the cellular sensing machinery undergoes and the associated increase in the tumbling frequency. The second reason is related to the steady decrease over time in the number of cellular trajectories identified by TaxisPy, which compromises the representativeness of the tumbling frequencies calculated and potentially increases the noise at later time points. This observation was consistent over all recorded videos (see Fig. S1). For consistency, the tumbling frequency for the control condition was also calculated over a period of 20 s.

3.2. Visualizing the adaptation process of the chemotactic machinery of E. coli

The two-component system controlling the chemotactic response of *E. coli* is able to adapt to external stimuli (Macnab and Koshland, 1972; Berg and Berg and Tedesco, 1975; Alon et al., 1999). Adaptation refers to the temporal process by which the cellular tumbling frequency returns to its pre-stimulation state. As a proof-of-concept and to demonstrate the utility and versatility of TaxisPy in elucidating this process, we stimulated an *E. coli* culture with 1 mM L-aspartate and videotaped its swimming pattern at regular time intervals for 40 min. Each video had a duration of 10 s. Constituting frames of each video were first extracted and subsequently analyzed using the functionalities provided by TaxisPy. The average tumbling frequency for each video was calculated using all trajectories detected in the whole duration of the video, i.e., 10 s. These data are graphically represented by the blue dots in Fig. 5. Black dots in the same figure represent the tumbling frequency of a control culture without the addition of the chemical attractant. The idea of the minimum average tumbling frequency used to analyze the chemotactic response of *P. putida* was not necessary here because the adaptation process of the chemotactic machinery of *E. coli* to L-aspartate takes several minutes. Additionally, temporal processes involved in the diffusion of the chemical attractant in the medium are not expected to be relevant in this case because the culture was kept in a shaking incubator at 200 rpm during the experiment.

The expected cellular adaptation behavior is evident from the time course of the tumbling frequency for the stimulated culture (blue solid line in Fig. 5). As time goes by, the average tumbling frequency increased from values lower than 0.05 tumbles/s during the first 10 min, to reach a maximum value of 0.25 tumbles/s after 20 min. The adaptation process can be characterized by two parameters: the precision of adaptation and the adaptation time (Alon et al., 1999). The first parameter is defined as the ratio between the average cellular tumbling frequency of the *unstimulated* culture and that of the *stimulated* culture after adaptation has been reached. The second parameter is defined as the time where the tumbling frequency of stimulated cells rises to halfway between its earliest measured value and its steady-state value. For the specific system under analysis, the adaptation time was 15 min and the precision of adaptation was 1.18.

4. Discussion

A major difficulty related to the customized use of cell tracking software is the determination of key parameter values for the identification of tumbles or changes of direction. In the past, such parameters

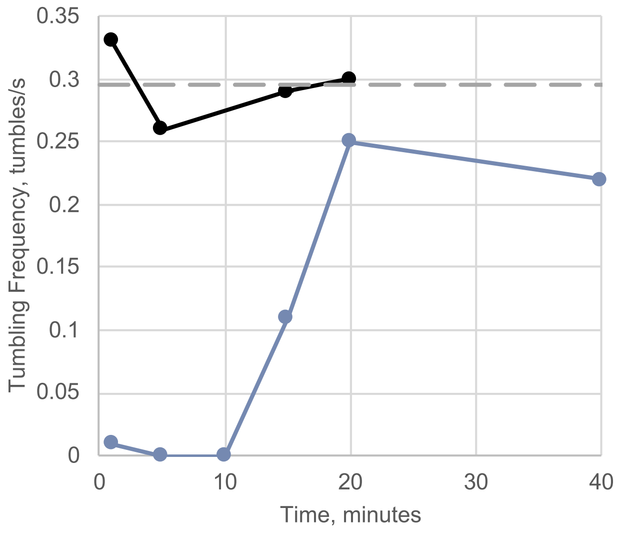


Fig. 5. Adaptation of the chemotactic machinery of *E. coli*. L-aspartate (1 mM) was added to an *E. coli* culture at time 0 and its swimming pattern videotaped for 10 s in regular time intervals over 40 min. Each video was subsequently analyzed using TaxisPy to estimate the temporal evolution of the tumbling frequency. These data points are represented by the blue dots. The tumbling frequency of a control culture without attractant is represented by the black dots. The grey dashed line represents the average cellular tumbling frequency of the unstimulated culture (0.295 tumbles/s). Each data point was generated from the analysis of 86 to 420 individual cellular trajectories. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

were empirically determined (Berg and Brown, 1972; Harwood et al., 1989), which compromises the applicability of those parameters under different conditions – e.g., different microorganisms or stimuli. TaxisPy systematically addresses this issue by implementing various strategies designed to guide the selection of relevant parameter values. For instance, TaxisPy employs a genetic algorithm along with a condition-specific training set to identify optimal values for three key parameters: *# Frames*, *# Smooth* and *Acc. Threshold.* The first two parameters affect the way that TaxisPy smooths cellular velocity data by calculating the average from a number of frames (dictated by *# Frames*) a certain number of times (determined by *# Smooth*). Noisy positional data tend to require more smoothing cycles. TaxisPy analyzes the time course of the absolute acceleration –which is calculated from the smoothed velocity data– to identify tumbling events. We use the premise that a cell decreases its linear velocity when tumbling and increases it again thereafter. Thus, a change of direction or tumble is characterized by two adjacent peaks in the time course of the absolute acceleration. To avoid identifying normal fluctuations in the velocity during smooth swimming as a tumble, the acceleration peaks are required to surpass a threshold value dictated by *Acc. Threshold*. Since smoothing the velocity data decreases the height of the peaks in the acceleration plot, high values for *# Frames* and *# Smooth* are usually accompanied by low values for *Acc. Threshold* and vice versa.

One of the motivations for the development of TaxisPy was to offer a simple method for the quantitative analysis of bacterial chemotaxis. Following the way paved by previous studies that implemented video-based analysis of bacterial chemotaxis, we decided to base our method on the analysis of cellular motion data for the identification of tumbling frequencies. This led us to identify the necessity of condition-specific values for some parameters –*# Frames*, *# Smooth* and *Acc. Threshold*– and to develop features in TaxisPy for their identification. Even though constructing training sets and identifying optimal parameter values usually requires a couple of minutes, this procedure can become tedious if a large number of videos needs to be processed. A promising approach for the “parameter-free” identification of tumbles is the analysis of cellular trajectories by multilayer neural networks (Lecun et al., 1998; Ciregan et al., 2012). These networks could be first trained on a large data set of cellular trajectories –each one of them labeled with the number of tumbles– to then be used on new trajectories to estimate their number of tumbles. An advantage of such an approach would be that the model parameterization would be done just once and could be then applied for various microorganisms under different conditions. Paving the way towards this new approach, we are providing an initial training set in the Supplementary File 1, which was manually extracted and curated from the *P. putida* experiments and consists of over 700 cellular trajectories, each one provided with the number of observed tumbles.

In this study, we used two different experimental set-ups to study the bacterial chemotactic response. For *P. putida*, the temporal response to shikimate pulses of varying concentration was measured on *microscope glass slides*. Using the minimum tumbling frequency during the first 20 s of each experiment, we were able to quantitatively capture the effect of the shikimate concentration on the swimming behavior. As expected, we observed a decrease in the tumbling frequency as the concentration of shikimate was increased (Fig. 4A). Increasing the shikimate concentration over 100 μM did not seem to have a further effect on the chemotactic response of *P. putida*. Additionally, values calculated for the tumbling frequency were within the same order of magnitude as previous reports (Harwood et al., 1989), which validates the computations performed by TaxisPy.

As mentioned in the results section and shown in Figs. 3B and S1, the number of trajectories identified by TaxisPy consistently decreased during the course of each experiment. This is potentially related to a temperature or light-dependent taxis away from the observation field of the microscope. An indirect support to this claim is provided by a report by Paster and Ryu (2008), in which a tactic response was observed in *E. coli* as a function of temperature gradients. The authors showed that at temperatures below 31 °C, the response to thermal stimuli is similar to the chemotactic response. However, at temperatures above 31 °C, some cells showed an inverted response, switching from warm- to cold-seeking behavior. Additional factors to be considered might be related to oxygen depletion and the accompanied overall loss of bacterial motility (Douarche et al., 2009) and with cellular migration towards the nearest air/water interface due to aerotaxis (Taylor, 1983) and away from the focus of the microscope. From a practical point of view, a steady decrease in the number of cellular trajectories over time has two implications. First, it suggests that the uncertainty of data extracted from adaptation experiments performed on glass slides will increase over time, since the number of trajectories used to calculate tumbling frequencies will continuously decrease. Second, it poses a biological constraint on the maximal length of the videos when the swimming pattern is observed on a glass slide. In line with these observations, we limited the analysis of each *P. putida* video to the first 20 s.

In the case of *E. coli*, a different experimental set-up was followed. Since the adaptation time exhibited by this microorganism under the studied conditions is in the order of magnitude of minutes, the actual adaptation process was conducted in an *Eppendorf tube* instead of on a microscope slide. Samples were taken from the vessel, which was kept under shaking conditions, and analyzed under the microscope for 10 s. In this way, we were able to estimate both the adaptation time and the precision of adaptation. The estimated value for the adaptation time of 15 min and the precision of adaptation of 1.18 are consistent with previously reported values (Alon et al., 1999). These results serve as further validation of our methods.

TaxisPy was designed for users with limited programming knowledge. Its user interface provides necessary tools to split a video into its constituting frames, identify cells in individual frames, link these to obtain trajectories, filter anomalous trajectories and calculate tumbling frequencies. Thanks to its convenient distribution through Docker, intuitive user interface and biologically feasible results, TaxisPy represents a valuable computational tool for the quantitative analysis of bacterial chemotaxis.

Declaration of Competing Interest

The authors declare no conflicts of interest.

Acknowledgements

This work was supported by grants from the National Science Foundation to R.E.P and M.A.S (MCB 1716833), and the USDA National Institute of Food and Agriculture, Hatch/Evans-Allen/McIntire Stennis project 1020219 to R.E.P.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.mimet.2020.105918.

References

Adler, J., 1969. Chemoreceptors in bacteria. Science 166, 1588–1597.

Adler, J., 1973. A method for measuring chemotaxis and use of the method to determine optimum conditions for chemotaxis by *Escherichia coli*. J. Gen. Microbiol. 74, 77–91.

Alexandre, G., 2010. Coupling metabolism and chemotaxis-dependent behaviours by energy taxis receptors. Microbiology 156, 2283–2293. https://doi.org/10.1099/mic.0.039214-0.

Allan, D., van der Wel, C., Keim, N., Caswell, T.A., Wieker, D., Verweij, R., Reid, C., Grueter, L., Ramos, K., Perry, R.W., Boulogne, F., Sinha, P., Bruot, N., Uieda, L., Katis, J., Mary, H., Ahmadia, A., 2019. Trackpy v0.4.2. Zenodo. https://doi.org/10.5281/zenodo.3492186.

Alon, U., Camarena, L., Surette, M.G., Aguera y Arcas, B., Liu, Y., Leibler, S., Stock, J.B., 1998. Response regulator output in bacterial chemotaxis. EMBO J. 17, 4238–4248.

Alon, U., Surette, M.G., Barkai, N., Leibler, S., 1999. Robustness in bacterial chemotaxis. Nature 397, 168–171.

Amsler, C.D., 1996. Use of computer-assisted motion analysis for quantitative measurements of swimming behavior in peritrichously flagellated bacteria. Anal. Biochem. 235, 20–25.

Beaulieu-Jones, B., Greene, C., 2017. Reproducibility of computational workflows is automated using continuous analysis. Nat. Biotechnol. 35, 342–346. https://doi.org/10.1038/nbt.3780.

Berg, H.C., Brown, D.A., 1972. Chemotaxis in *Escherichia coli* analysed by three-dimensional tracking. Nature 239, 500–504.

Berg, H.C., Tedesco, P., 1975. Transient response to chemotaxis stimuli in *Escherichia coli*. Proc. Natl. Acad. Sci. U. S. A. 72, 3235–3239.

Boettiger, C., 2015. An introduction to Docker for reproducible research. ACM SIGOPS Operating Syst. Rev. 49 (1). https://doi.org/10.1145/2723872.2723882.

Ciregan, D., Meier, U., Schmidhuber, J., 2012. Multi-column deep neural networks for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI. 2012. pp. 3642–3649. https://doi.org/10.1109/CVPR.2012.6248110.

Crocker, J.C., Grier, D.G., 1996. Methods of digital video microscopy for colloidal studies. J. Colloid Interface Sci. 179, 298–310. https://doi.org/10.1006/jcis.1996.0217.

Ditty, J.L., Parales, R.E., 2015. Protocols for the measurement of bacterial chemotaxis to hydrocarbons. In: McGenity, T., Timmis, K., Nogales, B. (Eds.), Hydrocarbon and Lipid Microbiology Protocols. Springer Protocols Handbooks. Springer, Berlin, Heidelberg.

Douarche, C., Buguin, A., Salman, H., Libchaber, A., 2009. *E. coli* and oxygen: a motility transition. Phys. Rev. Lett. 102,198101.

Drawert, B., Hellander, A., Bales, B., Banerjee, D., Bellesia, G., Daigle, B.J., Douglas, G., Gu, M., Gupta, A., Hellander, S., Horuk, C., Nath, D., Takkar, A., Wu, S., Lötstedt, P., Krintz, C., Petzold, L.R., 2016. Stochastic simulation service: bridging the gap between the computational expert and the biologist. PLoS Comput. Biol. 12, e1005220. https://doi.org/10.1371/journal.pcbi.1005220.

Finette, B.A., Subramanian, V., Gibson, D.T., 1984. Isolation and characterization of *Pseudomonas putida* PpF1 mutants defective in the toluene dioxygenase enzyme system. J. Bacteriol. 160, 1003–1009.

Fortin, F.A., De Rainville, F.M., Gardner, M.A., Parizeau, M., Gagné, C., 2012. DEAP: evolutionary algorithms made easy. J. Mach. Learn. Res. 13, 2171–2175.

Gibson, D.T., Hensley, M., Yoshioka, H., Mabry, T.J., 1970. Formation of (+)-*cis*-2,3-dihydroxy-1-methylcyclohexa-4,6-diene from toluene by *Pseudomonas putida*. Biochemistry 9, 1626–1630.

Harwood, C.S., Fosnaugh, K., Dispensa, M., 1989. Flagellation of *Pseudomonas putida* and analysis of its motile behavior. J.Bacteriol. 171, 4063–4066.

Hold-Geoffroy, Y., Gagnon, O., Parizeau, M., 2014. Once you SCOOP, no need to fork. In: Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment. ACM, New York, NY, pp. 60. https://doi.org/10.1145/2616498.2616565.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, MI.

Hung, L., Hu, J., Meiss, T., Ingersoll, A., Lloyd, W., Kristiyanto, D., Xiong, Y., Sobie, E., Yeung, K.A., 2019. Building containerized workflows using the BioDepot-Workflow-Builder. Cell Syst. 9, 508–514. e3. https://doi.org/10.1016/j.cels.2019.08.007.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86, 2278–2324.

Luu, R.A., Kootstra, J.D., Nesteryuk, V., Brunton, C.N., Parales, J.V., Ditty, J.L., Parales, R.E., 2015. Integration of chemotaxis, transport and catabolism in *Pseudomonas putida* and identification of the aromatic acid chemoreceptor PcaY. Mol. Microbiol. 96, 134–147.

Macnab, R.M., Koshland, D.E., 1972. The gradient sensing mechanism in bacterial chemotaxis. Proc. Natl. Acad. Sci. U. S. A. 69, 2509–2512.

Matilla, M.A., Krell, T., 2017. Chemoreceptor-based signal sensing. Curr. Opin. Biotechnol. 45, 8–14. https://doi.org/10.1016/j.copbio.2016.11.021.

Matilla, M.A., Krell, T., 2018. The effect of bacterial chemotaxis on host infection and pathogenicity. FEMS Microbiol. Rev. 42, 40–67. https://doi.org/10.1093/femsre/fux052.

McCall, J., 2005. Genetic algorithms for modelling and optimization. J. Comput. Appl. Math. 184, 205–222. https://doi.org/10.1016/j.cam.2004.07.034.

Parales, R.E., Ditty, J.L., Harwood, C.S., 2000. Toluene-degrading bacteria are chemotactic towards the environmental pollutants benzene, toluene, and trichloroethylene. Appl. Environ. Microbiol. 66 (4098–4014).

Parales, R.E., Luu, R.A., Hughes, J.G., Ditty, J.L., 2015. Bacterial chemotaxis to xenobiotic chemicals and naturally occurring analogs. Curr. Opin. Biotechnol. 33, 318–326. https://doi.org/10.1016/j.copbio.2015.03.017.

Parkinson, J.S., 1978. Complementation analysis and deletion mapping of *Escherichia coli* mutants defective in chemotaxis. J. Bacteriol. 135, 45–53.

Paster, E., Ryu, W.S., 2008. The thermal impulse response of *Escherichia coli*. Proc. Natl. Acad. Sci. U. S. A. 105, 5373–5377. https://doi.org/10.1073/pnas.0709903105.

Pfeffer, W., 1884. Locomotorische richtingsbewegungen durch chemische reize. *Untersuch. Bot. Inst.* Tübingen. 1, 363–482.

Pohl, O., Hintsche, M., Alirezaeizanjani, Z., Seyrich, M., Beta, C., Stark, H., 2017. Inferring the chemotactic strategy of *P. putida* and *E. coli* using modified Kramers-Moyal coefficients. PLoS Comput. Biol. 13, e1005329. https://doi.org/10.1371/journal. pcbi.1005329.

Sager, B.M., Sekelsky, J.J., Matsumura, P., Adler, J., 1988. Use of a computer to assay motility in bacteria. Anal. Biochem. 173, 271–277.

Sambrook, J., Fritch, E.F., Maniatis, T., 1989. Molecular Cloning: A Laboratory Manual, 2nd ed. Cold Spring Harbor Laboratory, Cold Spring Harbor.

Scharf, B.E., Hynes, M.F., Alexandre, G.M., 2016. Chemotaxis and signaling in model beneficial plant-bacteria associations. Plant Mol. Biol. 9, 549–559.

Stanier, R.Y., Palleroni, N.J., Doudoroff, M., 1966. The aerobic pseudomonads: a taxonomic study. J. Gen. Microbiol. 43, 159–271. https://doi.org/10.1099/00221287-43-2-159.

Staropoli, J.F., Alon, U., 2000. Computerized analysis of chemotaxis at different stages of bacterial growth. Biophys. J. 78, 513–519.

Taylor, B.L., 1983. How do bacteria find the optimal concentration of oxygen? Trends Biochem. Sci. 8, 438–441.

Tso, W.W., Adler, J., 1974. Negative chemotaxis in *Escherichia coli*. J. Bacteriol. 118, 560–576.

Valderrama-Gómez, M.A., Lomnitz, J.G., Fasani, R., Savageau, M.A., 2020. Mechanistic modeling of biochemical systems without a priori parameter values using the Design Space Toolbox v.3.0. iScience 23 (6), 101200. https://doi.org/10.1016/j.isci.2020.101200.

Webre, D.J., Wolanin, P.M., Stock, J.B., 2003. Bacterial chemotaxis. Curr. Biol. 13, R47–R49. https://doi.org/10.1016/s0960-9822(02)01424-0.