Characterizing Population Heterogeneity of *Salmonella* Motion in Mucosal Environments Using Stochastic Modeling and the EM Algorithm

by

Liane Solomon

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Applied Mathematics

Department of Mathematical and Statistical Sciences
University of Alberta

© Liane Solomon, 2022

Abstract

Salmonella are pathogenic bacteria that infect many species including humans. This pathogen thrives in the gastrointestinal track of their hosts and propel themselves in mucus with motion structures called flagella. Each cell has multiple flagella that can rotate either synchronously, resulting in directed motion, or asynchronously. This creates a distinct motion pattern known as "run and tumble" motion. A large particle tracking dataset of Salmonella in mucus harvested from mouse GI tract was recently published by Schroeder et al. [28]. This dataset includes a substantial fraction of cells that exhibit exclusively undirected motion while other cells exhibit exclusively run-and-tumble-type motion patterns. It is well known that *Salmonella* experience a significant amount of population heterogeneity in order to evade host immune cells, and this heterogeneity can manifest in motion patterns through its impact on flagella number and type. A systematic and quantitative statistical analysis of the Schroeder et al. dataset, informed by mechanistic stochastic models of cell motion, is performed to characterize motion heterogeneity. It is found that two distinct populations can be characterized that emerge from a rigorous statistical optimization procedure called Expectation Maximization. These two populations are described as "diffusers" and "swimmers." Interestingly, cells in the diffusers populations display random switching between distinct diffusivity values that differ by nearly 10 fold, indicating a previously unknown source of active motion among these otherwise non-motile cells. Approximately 8% of tracks were estimated to switch between these two subpopulations, most of which were attributable to tracking errors or uncertainty due to short track lengths. It is speculated that the remaining handful of tracks, which all transition from diffusers to swimmers, could be explained by young cells reaching the stage of development where they are able to generate directed motion.

Acknowledgements

I would like to thank the people who made the completion of this work possible. First and foremost, I'd like to thank my ever-patient supervisor Dr. Jay Newby, without whom I would not have a project. I'd like to thank the wonderful role models who showed me the beauty and magic in mathematics: my father Lloyd Solomon and grandfather Dr. Jimmy Solomon. I would like to thank the family, friends, and pets who offered me endless support and love through this process, specifically my mother Heidi, my sister Haley, my cats Pigeon and Magpie, and my pony Memphis. Finally, I would like to thank NSERC for providing the necessary funding for this project.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Salmonella holds a lot of name brand recognition for a simple bacterial species. This is perhaps because it is a highly infectious and common pathogen that most notably causes typhoid fever. Mainly thought of as a historical disease, as recently as 2004 there were 21-27 million cases world wide of typhoid fever, with up to 600,000 deaths annually. While typhoidal *Salmonella* is mostly found in South East Asia now, non-typhoidal *Salmonella* (NTS) can cause over 90 million infections annually, with as many as 12 million cases in the United States alone[20]. NTS is most commonly thought of as food poisoning, since food is a host for the pathogen. With antibacterial resistance rising, treatment and prevention of such infections is becoming a major, global concern. Beyond human infections, *Salmonella* can cause gastrointestinal diseases in most animals.

Upon entering the host body, *Salmonella* thrives and swims in mucosal conditions. The bacteria go on to penetrate and attach to epithelial cells along the gastrointestinal region. It has also been shown that *Salmonella* in-

vade host epithelial cells by creating ruffle like structures on the membrane of the host cells, and even hold specific preferences as to which areas to target when invading the epithelial cells.[1]. In terms of basic cell structure, it can be observed in Figure 1.1 that *Salmonella* cells are rod shaped bacteria with multiple flagella. The pathogenic nature of this species motivates an investigation of the physiology of *Salmonella* as well as an investigation into how they are propelled in mucosal environments, so that we can potentially combat the cells more effectively.

Figure 1.1: *Salmonella* cells in motion. The image on the left is a modification of an image from[24] and the image on the right is a modification of an image from[9].

Each *Salmonella* cell has multiple flagella, which when rotating synchronously create a directed motion force. If one flagella unsynchronizes from the rest, it disrupts the other flagella, causing the cell as a whole to experience undirected motion. This alternating motion pattern is referred to as run and tumble motion[25]. In order to examine this run and tumble motion pattern, Schroeder et al. harvested mucus samples from the GI tracts of mice[28]. These mucus samples hosted a plethora of *Salmonella* cells whose motion is recorded via microscopy videos. These videos, around $20 - 30$ seconds in length, were then translated into position data with the help of a particle tracking algorithm[28].

While Schroeder et al. expected to see a run and tumble motion pattern, they noticed some cells in their videos experienced low amplitude undirected motion. This prompted the development of a stochastic three-state motion model which included a dormant, run, and tumble state. Schroeder et al. hoped that this three-state model would answer questions they had about *Salmonella* motility and subsequently fit the model to their particle tracking data. They assumed that the cell were identical and switched between all states over the 30 second video time scale[28]. However, our own observations of these videos have led us to believe that another possible explanation for the motion patterns in our data could be the natural heterogeneity in *Salmonella* populations.

Any heterogeneity related to flagella could impact the motion *Salmonella* populations exhibit. Because *Salmonella* are pathogenic host immune pressures can lead to heterogeneity within single populations. Often, this heterogeneity is phenotypic adaptations to allow cells to avoid detection. One manifestation of this results in a fraction of cells within a population that do not develop or grow flagella[5, 31]. Even without host pressure, the number of flagella each cell has is variable. Additionally, the cell experiences different motion types as it develops flagella[17].

Through the use of the Expectation Maximization (EM) algorithm, we hope to develop and fit a biologically based stochastic process model for *Salmonella* motion that matches what we observe in the microscopy video from the Schroeder et al. data set. The goal is to to explore the alternative hypothesis that phenotypic heterogeneity is present in the population. We also seek to answer the following questions. Can we classify heterogeneous

sub-populations based on motion alone? Can we make predictions about the unobserved motion state (run, tumble, ect.) at time t? How do we fit parameters when the motion states at each time are not directly observed? What can we reveal about the underlying cause, or possibly the purpose, of these sub-populations based only on observations of motion?

One unique aspect of our approach to this project is creating a model that can describe two or more embedded sub-populations. Traditionally, when using an EM model for motion, the assumption is made that all objects follow the same motion model. However, two or more motion patterns are present under the heterogeneity hypothesis. Our approach for adapting the EM algorithm is based on the idea that one could embed two or more distinct models within a single larger model if the transition rate estimates result in disconnected subgraphs. Through the utilization of a (nearly) irreducible transition rate matrix, these motion patterns act as (almost) separate sub-models within the larger model.

This work will start in Chapter 2 with a biological investigation of *Salmonella* and their flagella. We will briefly look at previous models which attempt to capture the dynamics of *Salmonella* motion, as well as techniques used when examining tracking data. In Chapter 3, we discuss the mathematical and statistical tools. This specifically includes the Expectation Maximization algorithm, which is the parameter fitting tool we use for all models we test. Finally, in Chapter 4, we propose and examine several heterogeneous population models, using track visualizations and parameter estimations of each model. We also discuss the biological implications of our results, specifically in regard to population heterogeneity related to motion and motion structures.

Chapter 2

Background

2.1 Flagella Structure and Flagella Driven Motion

The first thing we will consider are the flagella, which drive *Salmonella* motion. Each flagella consists of four parts: a basal body, a C-ring, a rod-hook, and a filament. The structure of flagella can be seen in Figure 2.1. The flagella extends from the plasma membrane into the extracellular space. The basal body forms as a foundation in the plasma membrane, with the C-ring acting as an anchor. The filament is the structure we are most familiar with, as it is the helical polymer extending like a tail from the cell[32]. It is important to note that population heterogeneity can be found by examining the protein which makes up the flagella filament, as the tail can either be made of the FliC or the FljB protein, and any heterogeneity could impact motion patterns[31]. The rod-hook functions to attach the filament to the basal body, extending

through the cell envelop and into the extracellular space. The rod functions as an axle while the hook functions as a joint, allowing the filament to move in the propeller like fashion for which it is known[32].

The flagellar motor is actively responsible for propelling the cell forward. An ion gradient, called the ion motive force (IMF) powers the motors[22]. Experiments on *E. Coli*, another flagellated bacteria, have shown that transmembrane proteins, MotA and MotB play a strong role in the IMF generation for the bacterium.

One set of experiments called "motor resurrection" aimed to model flagellar torque generation the same way one would model electric motors[32]. The specifics of these experiments, first published by Blair and Berg in the paper, "Restoration of Torque in Defective Flagellar Motors," in Vol. 242 of *Science*, involve the repair of paralyzed MotA and MotB proteins with wild type proteins. The experiments indicate that these proteins play a large role in speed, force, and direction at which the motor spins, depending mainly on number of torque generators. The direction differences in rotation of flagella are particularly of interest to us, as this rotation causes the various types of movement a cell experiences[4]. Specifically, if all flagella rotate in the same direction synchronously, they generate forward, directed motion. However, if the rotation of one flagella unsychronizes from the rest, the directed force generated by that flagella counteracts the force generated by the other flagella, causing the cell to experience undirected motion.

Because of the importance of flagella and their relevance to bacterial motion, extensive studies have been conducted to understand them, even on a genetic level. In *Salmonella*, studies on the loci of flagellar genes and muta-

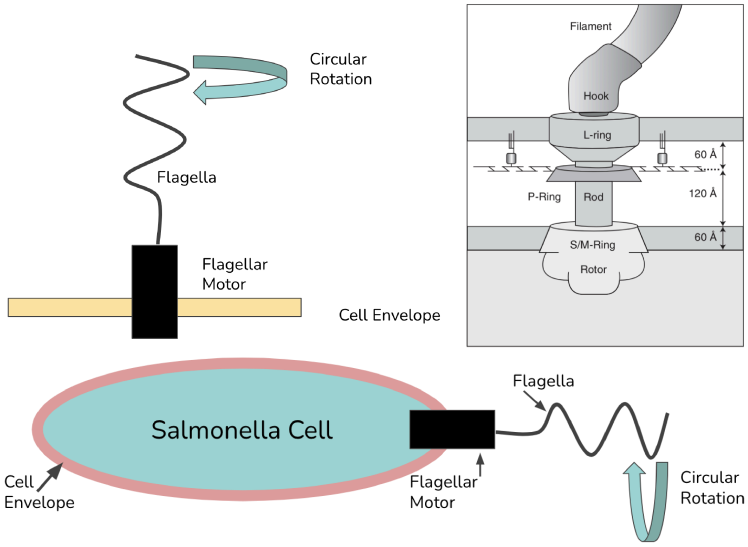

Figure 2.1: Basic Cartoons of flagellar Structure. Top left: macro view of whole flagellar structure. Top right: micro view of transmembrane parts of structure, image a modified version of one from Silhavy et al.[30]. Bottom: macro view of whole cell.

tions related to chemotaxis and motion are common. These studies led to the discovery of a family of genes with no homologous genes in *E. Coli*, a similar species of flagella driven bacteria. Interestingly, this genetic region has been shown to be responsible for a unique mechanism called phase variation. Phase variation means *Salmonella* posses two flagellar antigens which are genetically separate, but each cell can only express one at a time[19, 14]. However, the expression can switch to the other antigen. This leads to a level of inherent heterogeneity within an individual population of *Salmonella*. While the genetics of flagella certainly offers interesting insights, the assembly and mechanics of the structure answer other questions about cellular motion.

There are over 50 genes responsible for flagellar assembly, and these genes are typically considered in three classes of transcriptional units. Class one

is responsible for globally regulatory processes; class two is responsible for all proteins needed for the flagellar structure and assembly while class three regulates late stage assembly. It can observed that cellular motion such as rolling and swimming begins to occur as class three promoters become active. Additionally, the external tail of flagellar structures became visible with electron micrographs around this time[17]. This implicates flagellar assembly as a source of atypical motion patterns within populations of *Salmonella*, as the cells motion pattern actively changes during assembly.

2.2 Structural Features of the *Salmonella* Cellular Envelope

Another structure to consider is the bacterial cell envelope, as it often dictates a cells interaction with the environment. The multilayered bacterial cellular envelop not only offers protection for the cell but also allows nutrients in and waste out. *Salmonella* are gram negative bacteria, which means their cell envelope has three layers and contains lipopolysaccharide in the outermost layer. In contrast, gram positive cells only have 2 layers in their cellular envelope, but have surrounding layers of peptidogylcan. The structure of both gram positive and gram negative cells can be seen in Figure 2.2. As *Salmonella* are gram negative, we will take a closer look at the three layers of gram negative cell walls and their implications to the bacteria[30, 20].

The outermost layer of the cell envelope is the outer membrane (OM). Like many other cell membranes, the OM is made up of a lipid bilayer, however,

it is not a phoshoplipid bilayer like we expect in plant or animal cell membranes. The OM contains lipopolysaccharides (LPS) which are responsible for a number of interactions with extracellular stimulus, including interactions with host immune cells. The middle layer is the peptidogylcan cell wall, which gives bacteria its rigid structure. The inner most layer, the inner membrane (IM), is the traditional phoshoplipid bilayer membrane we expect. In bacteria, the IM hosts many of the proteins and protein structures that are functionally important, such as energy production, transport and secretion, and the synthesis of lipids.

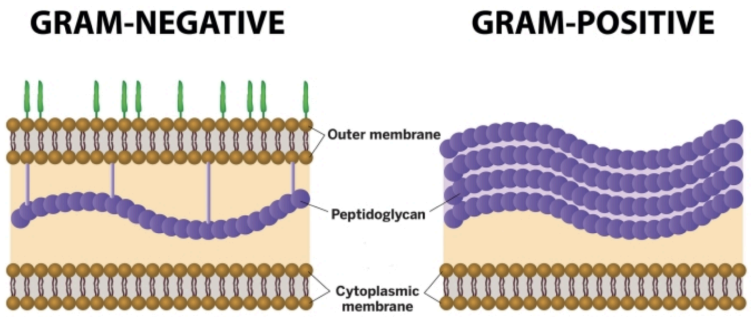

Figure 2.2: Structure of Gram Negative and Gram Positive Cells. This is a modified version of an image from Berg[2].

The triple layered cell membrane hosts a very important structure necessary for movement: the flagellar basal body hook structure. This is the structure that connects the filament to the flagellar motor, discussed in Subsection 2.1. In addition to the basic commonalities which all *Salmonella* share, these cells have a number of virulence factors which can differ within a population. This variation in virulence factors is what leads to the population heterogeneity we expect to see within our samples.

2.3 Population Heterogeneity

Due to the infectious nature of *Salmonella*, host immune cells produce external pressure to reduce populations of the bacteria. This pressure leads to well documented heterogeneity within the species as populations attempt to avoid detection and interactions with immune cells. It has been shown that there is number of virulence factors which can vary within a population. Everything from metabolic pathways to cell adhesion can demonstrate phenotypic heterogeneity in a population[5, 31, 33]. This includes the heterogeneity related to flagellar structure and assembly already mentioned in Section 2.1.

As flagella are what is responsible for the motion of *Salmonella* cells, we are most interested in heterogeneity surrounding these structures. One source of heterogeneity stems from the variability in the number of flagella typically found on single cells. This includes the fact that not all cells even develop flagella despite flagella being solely responsible for the motion of *Salmonella* and help with invasion and adhesion to host cells, making them pivotal to the cell's virulence. The importance of flagella speaks to the necessity of evading host immune cells, as some *Salmonella* opt to suppress flagellar assembly in order to evade the immune cells. While this is a very incomplete description of the heterogeneity present in a *Salmonella* population, it offers a strong motivation for considering model which accounts for a population with multiple, phenotypically heterogeneous sub-populations when investigating *Salmonella* motion.

2.4 Run and Tumble Models

In his 1976 lecture "Life at Low Reynolds Number" E.M. Purcell discusses "how microorganisms swim," specifically in low Reynolds number environments[25]. The Reynolds number is to the ratio of inertial forces to viscous forces in fluids. When viscous forces are dominant, we have low Reynolds numbers. Often, the environments in which microorganisms such as *Salmonella* thrive and swim are low Reynolds numbers environments because the microorganisms have a low mass, allowing the viscous forces to dominate over momentum. In this lecture, Purcell discusses not only the physical mechanics of how organisms in this environment swim and diffuse but also early experiments into imagining and tracking swimming microorganisms. One cited experiment conducted by Howard Berg tracks *E. Coli* cells in a variety of environments. Purcell describes the motion observed in these experiments as the cell "[swimming] for a while and then [stopping] and [going] off in another direction"[25]. This biphasic swim pattern is what will will discuss as run and tumble motion.

However, to understand run and tumble (RT) motion we must first understand random walk (RW) trajectories as RT motion leads to a RW trajectory over time. RWs are well studied stochastic processes in which position is changed by a random increment at every time step. For example, if we have a one dimensional random walk, we could explain it with the simple equation $x_{t+1} = x_t + \mu$, where μ is a random variable pulled from a well defined distribution, and x_t describes position at time t. Because of the random changes in position in RW, the agent experiencing RW trajectories often moves in random, different different directions, and models of RT motion emulate this.

RT models contain two phases that a particle can alternate between: the run phase and the tumble phase. The run phase consists of directed motion, often with a constant velocity. The tumble phase consists of the cell picking a new direction uniformly randomly, and the alternation between these motion types can be seen in Figure 2.3. The run phase is sometimes referred to as being "ballistic motion," however it can more easily be visualized as a random position change in a specific direction in a line with some constant velocity term[21]. This alternation is how RT models lead to RW trajectories over time.

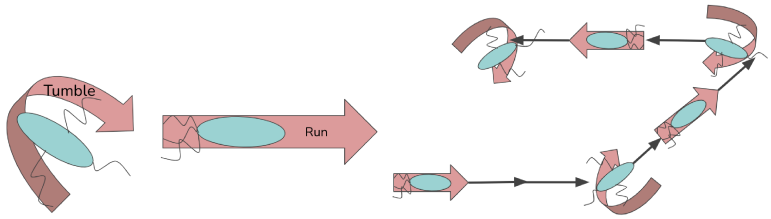

Figure 2.3: A cartoon demonstrating a run and tumble pattern.

RT models have been applied to many systems and the dynamics of such models have been extensively studied. One paper, "Run and Tumble Bacteria Slowly Approaching the Diffusive Regime," published in Physical Review in 2020, explores the long term dynamics of RT swimmers. The authors, Villa-Torrealba et al. were specifically after answers to questions about the time scale to reach diffusive regimes and system dynamics. By utilizing both theoretical methods and simulations, the researchers showed that mean-squared displacement, which is the measure of the deviation of position from a reference point, is an inaccurate measure of long term behavior, and alternative methods are needed to analyse such models. However, by examining the sharpness of the distribution of increments, Villa-Torrealba et al. classify the long term dy-

namics of run and tumble models with various swimming strategies[34]. While these results are mathematically motivated, other studies show the practical applicability of such models.

Just like *Salmonella*, *E. Coli* are flagella driven bacteria. Because of this, RT models can be used to explore *E. Coli* motion. Even models which build statistical simulations of *E. Coli* RT motion from mean squared displacement show significant accuracy. Miru Lee, Kai Szuttor, and Christian Holm outline one such simulation in their paper "A computational model for bacterial run-and-tumble motion". Their model incorporates hydrodynamics in order to more accurately model motion[21]. More commonly, Langevin equations are used to model run and tumble motion for swimming *E. Coli*. While the original Langevin equation describes simple Brownian motion, these equations now generally describe particle motion in fluid. Thus, Langevin equations represent the run or tumble states very naturally[8]. Another study by Bertrand et al. examine how such particles function in crowded environment[3].

Another practical application of run and tumble includes the model's relationship to chemotaxis. To investigate this relationship, a stochastic velocity-jump process, where an individual's velocity switches randomly, is used to model single cell movement. Unlike our model where the cell's motion is modeled through jumps in position, velocity-jump processes model motion through instantaneous changes in velocity. This allows for a transition from individual based run and tumble models into PDE models and cartoon based models for population level chemotactic movement with the Patlak-Keller-Segel equation given by

$$\frac{\partial n}{\partial t} = \nabla \cdot (D_n \nabla_n - \chi n \nabla S). \tag{2.1}$$

Additionally, Run and tumble models can be used to simulate numerically chemosensitive movement patterns based on Cattanelo's law which relates to heat propagation[7, 37, 38, 13, 39]. While these chemotaxis models are not as simple as pure run and tumble models, they build from run and tumble models and make use of the same biological principles. Run and tumble models are widely applicable and offer many insights into flagellated bacteria. Because of the relevance these models have, we will be expanding on run and tumble models as we explore our own models for *Salmonella* motion.

2.5 Experimental Data of *Salmonella* Motion

For this project, we are lucky to have abundant experimental data. This data set has already been used in published work by Schroeder et al[28]. The data comes in the form of microscopy videos from various mucosal conditions and locations in the gastrointestinal tract which have been converted into position data via particle tracking. The three main mucosal conditions are samples from MUC2 porcine GI tracts, $Rag^{-/-}$ mice GI tracks, and wild type mice GI tracks. Each set include samples from the duodenum, ileum, and jejunum. While wild type mice have no notable genetic mutations, $Rag^{-/-}$ mice lack mature B and T lymphocyte cells, reducing redundant antibodies which could interfere with the *Salmonella*[28]. MUC2 porcine have MUC2 mucin, which forms a net-like screen[15]. Within both the MUC2 and $Rag^{-/-}$ samples, there are additional conditions.

Both MUC2 and $Rag^{-/-}$ sets include samples which have added anti-lipopolysaccharide antibodies (anti-LPS IgG) and samples which have added

anti-biotin antibodies (anti-biotin IgG). The anti-LPS IgGs bind directly to the lipopolysaccharide in the outer membrane of the *Salmonella* cell, as introduced in Section 2.2. Part of the idea behind creating data samples of the anti-LPS IgGs is to investigate if these antibodies inhibit more motion than the anti-biotin IgGs, which act as a control group[28]. The wild type data set is made up of microscopy video of *Salmonella* extracted from the GI tract of wild type mice. That is, unlike the $Rag^{-/-}$ data set, there are no noticeable mutations in the mice that would impact the mocuosal environment of their gastrointestinal tracts. This data set is comprised of videos from the duodenum, ileum, and jejunum from a number of different mice.

The wild type data set contains tracks from three locations in the gastrointestinal tract, the duodenum, the ileum, and the jejunum. Overall the duodenum has $12,553$ data tracks, the ileum has $11,305$ data tracks, and the jejunum has $11,948$ data tracks. The position data is in the form of x and y position through time. The algorithm we use to fit our data to our models requires the data be in the form of increments that are the change in x and change in y position through time. Once the conversion to increment is made, we has 532,583 data points in the duodenum, 431,711 in the ileum, and 453,331 in the jejunum. That means we have over 1,417,625 change in position data points in the wild type data set alone. This is a stark comparison to the $Rag^{-/-}$ data set, which only has around $275,000$ data points, divided into multiple conditions. Figure 2.4 gives examples of stills from the microscopy videos.

As mentioned earlier, the videos were not sufficient data for the model proposed by Schroeder et al.; they needed position data. To convert the video

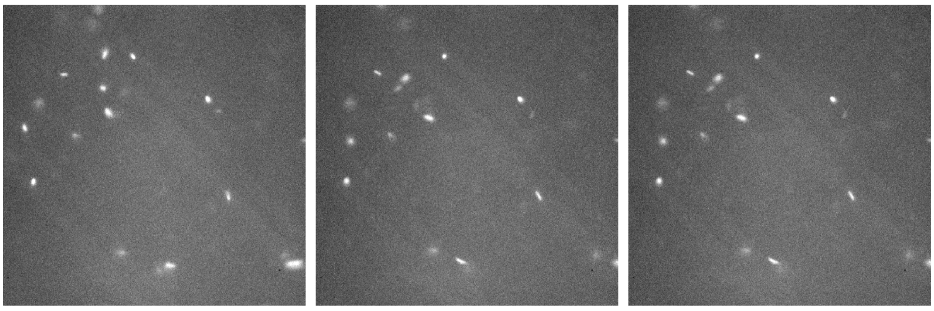

Figure 2.4: Pictured above are three images from the same microscopy video of wild type *Salmonella*. The images are in temporal order from left to right and taken 5 frames apart.

data to position data, they used a convolutional neural network (CNN) which localizes position data from images and videos. First formally proposed by Newby et al. in their 2018 paper "Convolutional Neural Networks Automate Detection for Tracking of Submicron-Scale Particles in 2D and 3D," the particular CNN we use has been tested on a multitude of 2D and 3D images across a variety of conditions. The CNN made similar tracking predictions to other methods, such as Mosaic. Utilizing three layers, the CNN that we use produces an x and y position for every particle at every frame, giving us what we need to calculate change in position data[23].

After being tracked, the data is in the format of (t, x, y) for every track saved into a CSV file, and all data points was of the form $(t, \Delta x, \Delta y)$. When the CNN was applied to the data set for Schroeder et al., it was applied to all data, including out wild type data. Therefore, we fortunately inherited the necessary position data we needed, and we just needed to convert it to the change in position format we needed to conduct the analysis. Each model presented in this paper was ran over all $1,417,625$ data points present in the wild type data set, which sometimes took over a day to compute. To

compensate, we tested the model for subsections of the data. For example, we divided the wild type data set into ileum, jejunum, and duodenum for each model to get results for those three sections prior to running it on the complete set of data.

It is also important to note that these cells move in 3D, but the tracking algorithm only records 2D position, due to the microscopy videos being 2D. The complications that arise from the 3D motion being restricted to 2D data is accounted for in our parameter fitting algorithm, discussed in Section 3.2.4. The tracking data thus allows us to start fitting our models to data and make the desired conclusions about salmonella motion, even though one dimension is missing.

2.5.1 Previous Work with the Homogeneous Population Model Model

Schroder et al. presented a three-state homogeneous model, which has a run, tumble, and dormant state[28]. This model is described fully in Section 3.1.4. By utilizing the data from the $Rag1^{-/-}$ mice, which lack mature T and B lymphocytes, the impact of anti-LPS immunoglobulins (IgG) on motion could be measured utilizing anti-biotin IgG as a control. Since *Salmonella* are gram negative bacteria, as discussed in Section 2.2, the anti-LPS IgGs could bind directly to the outer membrane of the bacteria. The paper on this work, subsequently published in Mucosal Immunology by Schroeder et al. in 2020, demonstrated that there was a difference in motion when the cells were exposed to the two different antibodies.

One of the main results found in this paper was that the percentage of time increments spent in the swimming state was reduced in the anti-LPS condition by about 10% from the anti-biotin condition. The model is also able to indicate a difference in velocity magnitude between the two conditions, with the anti-LPS having a slower velocity than the anti-biotin. This indicates that the cells in the anti-LPS condition experience more undirected motion. Beyond the fact that these results support the applicability of the three-state homogeneous model, they also indicate that mucosal conditions can impact *Salmonella* motion and flagellar rotation[28]. This work also demonstrates the applicability of a run and tumble model with an added dormant state, which is where will start for our exploration of the wild type data set.

We chose to work with the wild type set as opposed to the $Rag1^{-/-}$ set because of the size and consistency of the wild type data set, as the $Rag1^{-/-}$ set has only around $275,000$ data points divided into multiple conditions. While the $Rag1^{-/-}$ has a number of different condition sets which are interesting, having these condition sets did not add to our goal of creating a model which matches the natural movement patterns we can observe.

2.5.2 Previous Particle Tracking Studies of Bacterial Motion

Particle tracking is an important tool that is used in experiments studying multiple types of bacteria including rod shaped bacteria and lactic acid bacteria. However, a unifying goal of most experiments involving particle tracking is the goal to identify values such as first passage time or mean square displacement

in an attempt to determine the speed at which a cell moves through a given environment. Often, these experiments are centered around concepts such as infection speed of pathogens, drug dispersal, or other medical connections to microorganisms[40, 18, 27].

While our use of a particle tracking algorithm to generate our position data from microscopy videos is not unique, the number of data points we have is substantially greater than any previous study we are aware of. Even comparing the wild type set to the $Rag1^{-/-}$ set, which was used for Schroeder et al.'s paper, the wild type data set has over 5 times the number of time points. In comparison, Beljouw et al. a total of around $10,000$ tracks for their experimental lactic acid bacteria by-track data. The authors however only have 7 tracks over 40 frames, with the majority of tracks consisting of less than 20 frames[27]. Similarly, Bedrossian et al. claims to have between 6 and 149 cells in the field of view creating tracks for a total of 187 increments for their work on creating a new particle tracking algorithm using digital holographic microscopy[18]. While these are just a few specific examples, and there is a level of variability in the number of data points used, the fact that we have $1,417,625$ increments of data from $35,806$ tracks and 421 microscopy videos speaks to the depth of our experimental data.

Another more unique technique used for this project is a by-increment analysis of the particle tracking data. Often, when working with position data for microorganisms gathered with a particle tracking algorithm, a by-track analysis of the data is done[36, 27]. However, we chose to group all the data together and do a by-increment analysis because it can be shown that by-track analysis create biases in calculations that require averaging over

data[36]. These biases and the differences between by-track and by-increment analysis is explored in the next subsection, Subsection 2.5.3.

2.5.3 Tracks versus Increments

One important question when considering data from particle tracking is whether to consider the data by track or by increment. Due to the nature of particle tracking, there are two ways to consider the position data for a specific cell. Tracks, or traces, refer to the entire path of a tracked particle. However, this can lead an incomplete picture of what is happening as particles can move in and out of the plane of focus or field of view of the video. In general, faster moving particles are more likely to leave the field of view and produce shorter tracks. Additionally, if the particle moves a large distance between frames, the tracker will sometimes split the path into two separate traces, which is another problem for faster moving particles. Since we are investigating cells which can experience directed motion, this is an important consideration. Alternatively, one can parse the tracking data by increment, where each time point is considered individually. These two methods produce different results when calculating values that require averaging. The difference in these two approaches is the focus of the Wang et al. paper "Minimizing Biases Associated with Tracking Analysis of Submicron Particles in Heterogeneous Biological Fluids," and this difference was a consideration for us as we began our project[36].

Wang et al.'s paper sheds light on the bias created when a by-track averaging method is used during the investigation of tracked particle data with

heterogeneous movement patterns, that is movement patterns which appear as distinct. The authors discovered that if shorter time scales are used, the results favor the faster particles. The bias toward faster particles creates an issue when investigating biological processes such as the time it takes for a particle to cross a biological barrier. For example, this means this method could overestimate the speed at which a drug takes effect. In order to investigate these biases and the impact that a by-increment analysis has on the results, Wang and his team not only generated simulated data, but also collected biological data. They then compared the results of increment based analysis and track based analysis[36].

The authors calculated mean square displacement (MSD) with the formula $MSD = \langle|x(t) - x_0|^2\rangle + \langle|y(t) - y_0|^2\rangle$ where (x_0, y_0) is some initial position for the two dimensional simulations, as well as effective diffusion (D_e) by $MSD = 4D_e t$. The calculation of D_e was performed both by track and by frame. Before comparing the methods, the ground truth for the simulated data was established. The simulated data consisted of a mixture of particles with directed motion and particles with undirected motion with a 1:1 ratio of undirected motion particles to directed motion particles. Based off the simulations of homogeneously mixed fast and slow particles, both of these calculations should have distributions with no skew, or distortion in the distribution, toward either particle type for the by-track and by-increment calculation of D_e[36].

What the authors saw however was that in shorter time scales, there was a 85% skew in the distribution of D_e toward the faster particles with the by-track analysis. Similarly, the MSD for the by-track analysis on shorter time

scales was 10-fold higher than expected. This bias was verified with the in silco biological experiments. However, when the by-increment analysis was done on both the real and simulated data, the skews toward fast moving particles on the distribution of the D_e were within 1% of the predicted value. Additionally, the MSD was only 10% lower than expected[36]. The main take away from Wang et al.'s results is that the by-increment analysis of particle tracking data aligns more closely with the expected theoretical results.

To verify the results in Wang et al.'s paper, we also create simulated tracks which mirror the ones presented in the paper. Two groups of particles are created; one experiencing a fast rate of diffusion, and one experiencing a slow rate of diffusion. Just like in the paper, we vary the number of time points as well as the size of the time step. We pick a field size and if a particle leave the field, the data points are not counted. This leads to the fast particle creating a larger number of shorter tracks than the slow particle, which is expected. We then calculate effective diffusion (D_e) both by track and as well by increment to confirm the bias. These calculations are done for multiple rounds of simulations, with multiple step sizes and different numbers of steps.

Using the formula

$$D_e = \frac{\sum_{t=0}^{n} ||\Delta X_t||^2}{4n\Delta t},$$

which we derive by maximizing the likelihood function for the simulations and is described in more detail in Section 3.2.2, we are able to calculate the effective diffusion for n time points. When done by increment, the calculation of D_e looks identical to the given formula for D_e and the total time points includes all tracks. When done by track, the formula for D_e is applied to each track,

then D_e is calculated by averaging the by-track results, i.e.

$$D_e = \frac{\sum_T \frac{\sum_{t=0}^{n} ||\Delta X_t||^2}{4n\Delta t}}{T_{total}},$$

where $\sum_T$ is the sum over tracks and T_{total} is the total number of tracks. The results from our simulation using these calculations for effective diffusion then confirms the biases presented in Wang et al.'s paper. As we increase step size, D_e calculated by track approaches the ground truth diffusivity value for the slower population. Similarly, as we decrease step size, D_e calculated by track approaches the ground truth diffusivity value for the faster population. However, while those biases are still present when D_e is calculated by track, it is significantly reduced.

To demonstrate the difference between by-track and by-increment analysis, we created plots, shown in Figure 2.5. The figures demonstrate that the by-increment analysis estimate D_e more accurately than the by-track analysis, indicating that averaging by increment is a more accurate way to consider tracking data. Through an investigation of literature and our own independent simulations and analysis, we confirmed that we need to conduct all of our analysis averaging by increment. This means, whenever we have to sum the data points and normalize, it must be done for the entire data set, not just for a specific track. This shift from the traditional by-track analysis for tracked particles reduces bias in populations where not every particle is moving at the same speed, and some particles are moving quickly, which we see in our wild type data set.

To create a mathematically and biologically motivated model for *Salmonella*

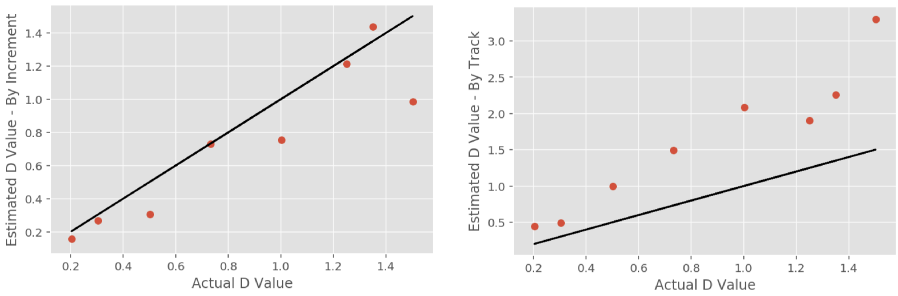

Figure 2.5: The accuracy of by-track averaging vs by-increment averaging is demonstrated by plotting the actual D_e value verses the estimated D_e value. The line is the $x = y$ line to show how close to the ground truth the estimations are. Left is the by-increment averaging and right is the by-track averaging

movement in mucosal conditions we will not only consider everything we have laid out in this chapter, but we will also need to consider mathematical methods. This will allow us to be precise and gather specific results in order to answer some of the questions we have about our data set including the nature of observable heterogeneity present. These mathematical methods will influence the structure and function of our models, as well as demonstrate how results can be generated.

Chapter 3

Mathematical, Statistical, and Computational Tools

3.1 Stochastic Models of Salmonella Motion

As *Salmonella* rotate their flagella, multiple sources of random forces impact their motion. These forces include random collisions with other molecules. Additionally, a cell can randomly change how and if each flagella rotates, creating a stochastic switch of states. Because of this inherent stochasticity, it is important for us to look at the nature of stochastic processes as the behavior of the stochastic processes we use will inform our analysis of the bacterial behavior. A stochastic process is a function of a time dependent random variable[16].

One specific type of stochastic process is a Markov process. Markov processes all have the Markov property, which means the probability of seeing a specific observation at integer valued time t only depends on time $t-1$ and

no other proceeding time points. In other words, if we have the value x_{t-1} at time $t-1$, the conditional probability $P(x_n, t_n|x_1, t_1; ...; x_{t-1}, t_{n-1})$ can be reduced as

$$P(x_n, t_n|x_1, t_1; ...; x_{t-1}, t_{n-1}) = P(x_n, t_n|x_{t-1}, t_{n-1}), \tag{3.1}$$

for $t_1 < t_2 < t_3$. We can also decompose the joint probability $P(x_1, t_1; x_2, t_2; x_3, t_3)$ in the following way

$$P(x_1, t_1; x_2, t_2; x_3, t_3) = P(x_1, t_1; x_2, t_2)P(x_3, t_3|x_1, t_1; x_2, t_2)$$

$$= P(x_1, t_1)P(x_2, t_2|x_1, t_1)P(x_3, t_3|x_2, t_2). \tag{3.2}$$

This indicates that $P(x_1, t_1; x_2, t_2; x_3, t_3)$ can be calculated completely from $P(x_1, t_1)P(x_2, t_2|x_1, t_1)$. Additionally, one can integrate equation 3.2 over x_2 in order to derive the Chapman-Kolmogorov equation in the form

$$P(x_3, t_3|x_1, t_1) = \int_{-\infty}^{\infty} P(x_2, t_2|x_1, t_1)P(x_3, t_3|x_2, t_2)dx_2. \tag{3.3}$$

This equation indicated that because of the Markov property, the probability of going from x_1 to x_3 is equivalent to going from x_1 to x_2 and then x_2 to x_3, summing over all possible x_2[16].

When a Markov process is stationary, that is $P(x, t)$ does not depend on t and $P(X_i, t_i|x_j, t_j)$ depends only on $\tau_{ij} = t_i - t_j$, this derivation simplifies further. In this case, the transition probability P only depends on the time step size, $\tau_{ij} = t_i - t_j$. If we call $P(x_2, t_2|x_1, t_1) = T_\tau(x_2|x_1)$, we see equation

3.3 turn into

$$T_{\tau+\tau'}(x_3|x_1) = \int T_\tau(x_2|x_1)T_{\tau'}(x_3|x_2)dx_2, \tag{3.4}$$

for τ, τ' positive[16]. As the stochastic processes we work with in this paper are all stationary Markov processes, we can consider this second version of the Chapman-Kolmogorov equation given by equation 3.4.

3.1.1 The Master Equation and Other Products of the Chapman-Kolmogorov Equation

The deviation for the Chapman-Kolmogorov equation given by 3.4 can be represented in a different form for Markov processes, known as the master equation, given vanishing time steps τ', and for simplicity, we will be using $P(x,t|x_0,t_0) = T_{t-t_0}(x|x_0) = P(x,t)$. This equation is given by

$$\frac{\partial P(x,t)}{\partial t} = \int [Q(x|x')P(x',t) - Q(x'|x)P(x,t)]dx', \tag{3.5}$$

and the solution of this differential equation for a given initial condition is $T_{t-t_1}(x|x_1)$ and given a jump at time t, $Q(x|x')$ is the transition probability from x' to x[16, 10]. That is, the transition probability for the Markov process. A complete derivation of the master equation from 3.4 can be viewed in N. G. Van Kampen's book "Stochastic Processes in Physics and Chemistry"[16].

Another important product of the Chapman-Kolmogorov equation occurs when we consider a discrete state set, as in the case of all models discussed in this paper. We can then re-write the master equation as given in equation

3.5. If we have n states, we get

$$\frac{\partial P_i}{\partial t} = \sum_{j=1}^{n} [Q_{ij} P_j(t) - Q_{ji} P_i(t)], \tag{3.6}$$

which is notionally equivalent to

$$\frac{\partial P(t)}{\partial t} = W P(t), \tag{3.7}$$

where $P_i : [0, \infty) \to [0, \infty)$, making P a vector valued function with n function entries. This is an important form of the master equation because W in this equation is known as the transition rate matrix and is a pivotal part of the solution to the differential equation, $P(t) = e^{tW} P(0)$, given initial condition $P(0)$ [16]. Each element, x_{ij} of W represents the rate of transitioning from the state i to the stater j. Additionally, part of this solution, $\Phi = e^{tW}$ is what is called the matrix exponential, and will be a fundamental part of our Expectation Maximization algorithm. W and the matrix exponential are important values to consider when working with discreet Markov processes.

In equation 3.7, W is a matrix comprised of transition rates and has a specific structure. An important property of W is irreducibly. An irreducible transition rate matrix allows for the potential for transitions through all the states. That is, an actor in a stochastic process with an irreducible transition rate matrix has the potential to take on any state value in finite time. If a transition rate matrix is reducible however, it is not possible for an actor to transition through all states. A reducible matrix is a matrix which can be put in the form of block upper triangular matrix through row and column

permutations. That is, a matrix which can be reduced to the form

$$W = \begin{bmatrix} C_{11} & C_{12} \\ 0 & C_{22} \end{bmatrix},$$

where C_{11}, C_{12}, and C_{22} are blocks of irreducible sub-matrices, is a reducible matrix[16]. We will consider stochastic processes with both reducible and irreducible transition rate matrices in this paper.

The last variation to the master equation we must consider is if we send the quantities for Q in equation 3.5 to zero. Then, we can derive the Fokker-Planck equation given by

$$\frac{\partial p(x,t|x',t')}{\partial t} = -\sum_i \frac{\partial}{\partial X_i}[A_i(x,t|x',t')] + \frac{1}{2}\sum_{i,j} \frac{\partial^2}{\partial X_i \partial x_j}[B_{ij}(x,t)p(x,t|x',t')], \tag{3.8}$$

where A is the drift vector and B is the diffusion matrix, making the whole equation representative of a diffusion process. From equation 3.8, we can develop SDEs for diffusion processes, given that A is a velocity and we superimpose a Gaussian fluctuation onto the covariance matrix B. This SDE would take the form

$$dx(t) = A(x(t),t)dt + B(x(t),t)N(0,dt), \tag{3.9}$$

where $N(0,dt)$ is a Gaussian random variable with mean zero and variance dt. A full derivation of this can be viewed in Gardiner's book "Handbook Of Stochastic Methods for Physics, Chemistry, and the Natural Sciences"[10]. For this project, all of the SDEs we use will be of the form given in equation 3.9.

3.1.2 Long Term Behavior of Stochastic Processes

When considering differential equations, such as the master equation, one important aspect to consider is the long term behavior of the solutions. To do this, we must set the derivative equal to zero. In this case, we can consider

$$0 = WP(t), \tag{3.10}$$

given initial condition $P(0)$. As this is a linear system, the structure of W plays an important role in the existence and uniqueness of a steady state solution $P_\infty = \lim_{t\to\infty} P(t)$. If we first consider the case of an irreducible transition rate matrix, it can be shown that as $t \to \infty$, a unique steady state solution, P_∞, exists such that $P(t) \to P_\infty$ for any valid $P(0)$. However, in the case of a reducible transition rate matrix, long term behavior is slightly more complicated. It can be shown that for a reducible transition rate matrix, more than one steady state solutions might exist, but for n finite state there will always be convergence to one of these solutions, depending on initial conditions. Detailed proofs of the long term behavior of stochastic processes can be found in Van Kampen's book "Stochastic Processes in Physics and Chemistry" [16].

Asymptotic mean velocity is another indicator of long term behavior that we can consider. Given by the equation

$$V_\infty = \lim_{t\to\infty} E[\frac{X(t) - X_0}{t}], \tag{3.11}$$

for some initial condition X_0, asymptotic mean velocity is the expected value of the change in position over all time, in other words the expected value of

overall velocity. We define directed motion as a process where $V_\infty \neq 0$.

The three-state model, given by equations 3.13 and 3.14, introduced by Schroeder et al. and used in this project is a "doubly stochastic" process. Stochastic state switches exist between swim, tumble, and dormant, and each state contains a stochastic process within itself. This adds a certain level of complexity, and to capture this, we can use transition rate diagrams to visualize models, for an example see Figure 3.1. Transition rate diagrams give a visual representation of the stochastic states, as the circles on the diagrams represent the states, as well as the stochastic transitions between states, represented by the arrows between the circles. Thinking about our models as doubly stochastic allow us to use specific statistical tools for parameter estimations. Additionally, the translation of stochastic processes into different forms in order to derive transition rate matrices, see equation 3.7 for more details, and SDEs, see equation 3.9, is a concept used later in this paper when considering our models.

3.1.3 Gillespie Algorithm

Daniel Gillespie first proposed his algorithm for simulating discrete stochastic processes in his paper "Exact Stochastic Simulations of Coupled Chemical Reactions"[12]. As the title implies, Gillespie aims to model coupled chemical reactions, in which molecular population levels can only take on integer values. The general idea of the algorithm is to stochastically simulate processes where there are changing whole number valued variables in a continuous way. To do this, the algorithm makes use of the fact that the value in question, in

Gillespie's case the molecules, are uniformly randomly distributed throughout the area. As Gillespie was interested in modeling coupled chemical reactions, it can be said that he was interested in the event of molecular collisions. Because the molecules are uniformly randomly distributed, as long as we know the event rate we can calculate when the next event will occur and what kind of event it will be[12].

To execute the Gillespie Algorithm, a few calculations and two random numbers are needed. First, given a_i, the event rate for event i, we must calculate $A = \sum_i a_i$ for all possible events. Then, two random variables, r_1 and r_2 are pulled from the unit uniform distribution. These two variables will subsequently be used in the following calculations:

1. $\tau = \frac{\ln \frac{1}{r_1}}{A}$

2. $\mu =$ event i such that $\sum_{k=1}^{\mu-1} a_k < r_2 A \leq \sum_{k=1}^{\mu} a_k$

With τ and μ, the simulation can then be propagated forward through time, as τ represents the time before the next event happens while μ indicates which event. That is, if the current time is t, event μ will happen at time $t + \tau$. We can generate our random numbers in this way because it can be proven that any two random numbers from the unit uniform distribution can create a random pair from any paired probability density function in this case r_1 and r_2 are mapped to τ and μ. In other words, if a random variable X has cumulative probability distribution $P(x)$ such that

$$P(x) = \text{Prob}[X < x] = \int_{-\infty}^{x} p(x')dx',$$

and random variable U has the unit uniform distribution, then $X = P^{-1}(U)$. In his original outline of the Gillespie algorithm, Gillespie discusses the reason for this translation with more rigor[12].

The Gillespie algorithm as described above can be utilized to simulate data for our model by simulating when state switches occur as well as the order of state switches. In order to calculate the position data needed to validate our EM algorithm, some additional modifications are needed. Specifically, between state switches, the particle will experience movement based off the state assignment.

3.1.4 A Homogeneous Population Model

Since *Salmonella* use flagella to swim, with distinctly different modes of swimming, a simple model can be created to represent *Salmonella* swimming in mucus. The SDE model presented in this section represents the cell's change in position and is the same biologically based model used by Schroeder et al.[28]. This model considers that a cell can experience undirected motion by including the dormant state to represent extended periods of undirected motion. Additionally, *Salmonella* also experience directed motion, or motion with a non zero mean velocity, fueled by the synchronized rotation of its flagella.

The model has three-states, dormant, tumble, and swim. While the state of motion, S, is fixed, an SDE expresses the change in position dependent on the state, S, $dX = v(\phi_t, \theta_t, S_t)dt + \sqrt{2D(S_t)}dW$, where $dW = N(0, dt)$ and $v(\phi_t, \theta_t, S_t)$ is the velocity of directed motion in state S with orientation ϕ_t and θ_t on the unit sphere. The two dimensional vector X is the change in position

in the x and y planes. Therefore,

$$v(\phi, \theta, S_t) = \delta_{S_t,2} \begin{bmatrix} v_{mag} \sin(\theta) \cos(\phi) \\ v_{mag} \sin(\theta) \sin(\phi) \\ v_{mag} \cos\theta \end{bmatrix}, \tag{3.12}$$

where v_{mag} is the velocity magnitude observed in all directions. Given that we are dealing with 3D motion, the motion of the cells is considered on the half unit sphere instead of the unit circle. We only consider the half unit sphere because Because we only have 2D data though, we only need to consider the x and y components of motion in these equations, as our EM code already accounts for the lack of data from the third dimension, as mentioned in Section 3.3.2. The fact that we have one velocity magnitude indicates a cell experiences the same velocity magnitude regardless of direction. While this seems like a strong assumption to make, consistent velocity regardless of direction is a biologically based assumption. In the wild type data set we have a homogeneous fluid environment and a dilute population, meaning the cell will move with the same constant velocity if it is swimming regardless of direction as no direction will offer significant hindrances. This model, which can be viewed in Figure 3.1, has the following master equation for the state transitions

$$\frac{\partial P(t)}{\partial t} = \begin{bmatrix} -k_1 & k_2 & 0 \\ k_1 & -k_2 - k_3 & k_4 \\ 0 & k_3 & -k_4 \end{bmatrix} P(t). \tag{3.13}$$

Let $S(t)$ be a sample from the distribution defined above. Then, we have the

following SDEs governing the position,

$$dX = \begin{cases} \sqrt{2D_0}dW, & S(t) = 0 \\ \sqrt{2D_1}dW, & S(t) = 1 \\ v(\phi_t, \theta_t)dt + \sqrt{2D_2}dW, & S(t) = 2. \end{cases} \tag{3.14}$$

Finally, the swim state corresponds to when the cell has forward directed motion. However, there are infinitely many directions the cell could take which would not allow for a discrete number of states. Therefore, we discretize the half unit sphere. This means that the swim state is not one state. If there are n polar angles and m angles of depression, that generates nm directions and thus nm swim states. Therefore, for a given ϕ_i and θ_j the SDE for swim state, s_{ij}, can be written in a discretized form as ($S = 2$):

$$dX = v(\phi_i, \theta_j, s_{ij})dt + \sqrt{2D_2}dW.$$

While similar to the dormant and tumble states, the swim state includes a non zero mean.

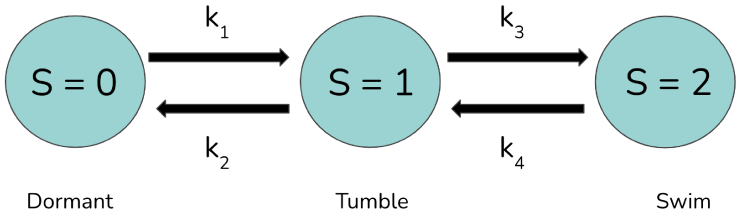

Figure 3.1: Model Diagram for three-state Model

One important tenant of the model is the restricted transition rates. As shown by Figure 3.1, a cell cannot move from dormant to swim, or from one

swim direction to another direction, without going through the tumble state first. This is because the cell uses the tumble state to reorient itself. Therefore, the number of transition rates is greatly reduced. If we have n swim directions, without unrestricted rates we would have $(n+2)^2$ distinct transition rates. However, with the restricted rates, there is only 4 possible transitions as seen in Figure 3.1. The four transitions are dormant to tumble, tumble to dormant, swim to tumble, and tumble to swim. While a cell cannot move directly from dormant to swim, a single track can have a time step in dormant and a time step in swim, making this transition rate matrix irreducible.

One assumption made about these transition rates is that a cell has an equal chance of reorienting in any direction during the tumble phase. That is, there is no bias in which run direction a tumbling cell transitions into. This assumption can stand because bacteria only have directional preferences when exposed to chemical signals, and there is no chemotaxis in our data sample. Additionally, our data is for a diluted population, which means it is highly unusual for the cells to interact and influence the motion of other cells. As the model is a procedure model for a very specific biological process, the constraints of restricted transition rates stands as a reasonable assumption.

This three-state homogeneous population model has already been applied to experimental data to demonstrate that mucosal conditions can impact motion, as seen in the paper by Schroeder et al[28]. As mentioned earlier in Section 2.5, position data for wild type *Salmonella* has been collected from *Salmonella* extracted from the gastrointestinal tract of various mice via microscopy videos and a particle tracking algorithm. Schroeder et al. took a specific portion of this data and examined their specific questions using this

model[28]. The conditions that data exists for and the results already found, including the impact of antibodies on motion, are explored in Section 2.5.1. We apply the EM algorithm for this model to the wild type data set in order to see if it captures the observed motion patterns and populations with heterogeneous movement patterns.

3.2 An Introduction to the Statistical Tools used for Parameter Estimation

The main statistical tool we use for this project is Expectation Maximization (EM). EM is used as a parameter estimation algorithm in the case of incomplete data, for example with Hidden Markov Models (HMMs). In particular, the EM algorithm can produce probability distributions which predict things like most probable state at a specific time. This sort of distribution could be useful to us as we attempt to characterize *Salmonella* motion. However, before we can understand EM algorithm, we must first take a look at likelihood functions and maximum likelihood estimation, as these are fundamental building blocks for the more complex Expectation Maximization algorithm.

3.2.1 Likelihood Functions

We can construct a function to describe the probability of seeing a certain set of observations, $\{X_t\}$, given the parameters, θ. This function, called the likelihood function holds a central place in the parameter estimations for stochastic models as this function describes the relationship between what we know and

what we want to explore. The likelihood function is a function of the parameters present in a system, not the samples, which allows us to treat the random variables as observations which are fixed[11, 26]. In the case of complete data, the set θ contains all of the variables in the network, and the likelihood function can be written as

$$L(\theta) = \prod_{t=1}^{T} P(X_t|\theta), \tag{3.15}$$

where $P(X_t|\theta)$ is the probability density function (PDF) that corresponds to the system. A probability density function is the probability that a value, X would be chosen when sampling that random variable, that is

$$\text{Prob}[X \in A] = \int_A P(x \mid \theta)dx.$$

where A is a set of interest that values X can take [29]. If one maximizes the likelihood function, one finds the set of parameters which most likely created the observations X_t. However, maximizing the likelihood function itself can be quite complex, so we instead consider the logarithm of the likelihood function.

If we assume our probability density functions are positive. Then, the likelihood function will be positive as it is the product of the probability density functions at every time point. Because the logarithmic function is concave and strictly increasing, optimizing the log of the likelihood function is equivalent to optimizing the likelihood function.

Lemma 3.2.1. *Maximizing the logarithm of a function that maps values from* $\mathbb{R}^n \to (0, \infty)$ *is equivalent to maximizing that function*

Proof. Let $g : (0, \infty) \to \mathbb{R}$ such that $g(x) = \ln(x)$, then $g'(x) = \frac{1}{x} > 0 \ \forall$

$x > 0$. Let $f : \mathbb{R}^n \to (0, \infty)$ be a differentiable function. If we consider the composite function $h(x) = g(f(x))$, $h : \mathbb{R}^n \to \mathbb{R}$, the chain rule tells us that

$$\nabla h(x) = g'(f(x))\nabla f.$$

Additionally, since $g'(f(x)) > 0$ it follows that $\nabla h(x) = 0 \iff \nabla f = 0$. Therefore, we see that maximizing the log likelihood also maximizes the likelihood. □

The process of maximizing the log likelihood can be used to estimate parameter values. This method of parameter estimation, referred to as Maximum Likelihood Estimation, is a foundational method for parameter estimation when working with stochastic processes. The specifics of Maximum Likelihood Estimation are explored in Subsection 3.2.2.

3.2.2 Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) obtains the parameters in a given model by optimising the likelihood or the log likelihood. When we have a complete data set for a given model, MLE offers an accurate estimation of the parameters through this method. For example, if we have basic Brownian Motion, $\Delta X = \sqrt{2D}\Delta W$ with ΔW being a normal random variable with mean zero and variance Δt, $\Delta W = N(0, \Delta t)$, for some Δt and we want to approximate D, we can consider

$$L(D) = \prod_{t=1}^{T} P(\Delta X_t | D). \tag{3.16}$$

Since we have a normal distribution for our random variable and $\theta = D$, we can use the Gaussian PDF as the likelihood function to get

$$L(D) = \prod_{t=1}^{T} \frac{1}{\sqrt{4\pi D\Delta t}} e^{\frac{-\Delta X_t^2}{4D\Delta t}},$$

$$\ln(L(D)) = \sum_{t=1}^{T} \ln(\frac{1}{\sqrt{4\pi D\Delta t}} e^{\frac{-\Delta X_t^2}{4D\Delta t}}),$$

$$\ln(L(D)) = \sum_{t=1}^{T} \ln(\frac{1}{\sqrt{4\pi D\Delta t}}) - \sum_{t=1}^{T} \frac{\Delta X_t^2}{4D\Delta t}.$$

This function can subsequently be optimized with respect to D by using logarithmic differentiation in one dimension to obtain

$$D = \frac{\sum_{t=1}^{T} \Delta X_t^2}{2T\Delta t}. \tag{3.17}$$

As additional complexities, such as switching diffusion or multiple dimensions are added, the formula becomes more complex as well. In the case where we need to solve for diffusion and velocity magnitude in the following example of Brownian motion

$$\Delta X = v_{mag} v(\phi_t, \theta_t)\Delta t + \sqrt{2D}\Delta W, \tag{3.18}$$

we have a bi-variant normal distribution where there is no correlation between the x and y positions. Thus, the probability density function can be written as

$$f(\Delta x, \Delta y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{\frac{-1}{2}[(\frac{\Delta x-\mu_x}{\sigma_x})^2 + (\frac{\Delta y-\mu_y}{\sigma_y}])^2}, \tag{3.19}$$

Where $\sigma_x = \sigma_y = \sqrt{2D\Delta t}$ and $\mu_x = v_{mag}v_x(\phi_t, \theta_t)\Delta t$ and $\mu_y = v_{mag}v_y(\phi_t, \theta_t)\Delta t$ where $v_x(\phi_t, \theta_t)$ is the velocity component in x, $v_y(\phi_t, \theta_t)$ is the velocity component in y, and v_{mag} is the velocity magnitude. For simplicity,

$$v(\phi_t, \theta_t) = \begin{bmatrix} v_x(\phi_t, \theta_t) \\ v_y(\phi_t, \theta_t) \end{bmatrix}.$$

Simplifying the exponent we see

$$\frac{-1}{2}[\frac{\Delta x - \mu_x}{\sigma_x} + \frac{\Delta y - \mu_y}{\sigma_y}] = \frac{-1}{2}[\frac{(\Delta x - v_{mag}v_x(\phi_t, \theta_t)\Delta t)^2 + (\Delta y - v_{mag}v_y(\phi_t, \theta_t)\Delta t)^2}{2D\Delta t}].$$

Therefore, the likelihood function can be written as

$$L(\theta) = \prod_{t=1}^{T} \frac{1}{4\pi D\Delta t} e^{\frac{-1}{2}\frac{(\Delta X_t - v_{mag}v_x(\phi_t,\theta_t)\Delta t)^2 + (\Delta Y_t - v_{mag}v_y(\phi_t,\theta_t)\Delta t)^2}{2D\Delta t}}. \tag{3.20}$$

Note here that $\theta = (v_{mag}, D)$, so this is a multivariate optimization problem, and we must take the derivative with respect to v and with respect to D. The subsequent log-likelihood function is

$$\ln(L(\theta)) = \sum_{t=1}^{T} \frac{1}{4\pi D\Delta t} + \sum_{t=1}^{T} \frac{-1}{2} \frac{(\Delta X_t - v_{mag}v_x(\phi_t, \theta_t)\Delta t)^2 + (\Delta y_i - v_{mag}v_y(\phi_t, \theta_t)\Delta t)^2}{2D\Delta t}$$

$$\ln(L(\theta)) = \sum_{t=1}^{T} \frac{1}{4\pi D\Delta t} + \sum_{t=1}^{T} \frac{-1}{2} \frac{||\Delta X_t - v_{mag}v(\phi_t, \theta_t)\Delta t||_2^2}{2D\Delta t}. \tag{3.21}$$

Differentiating with respect to diffusion, D, we see that

$$\frac{\frac{\partial L(\theta)}{\partial D}}{\ln(L(\theta))} = \sum_{t=1}^{T} \frac{-1}{D} + \sum_{t=1}^{T} \frac{||\Delta X_t - v_{mag}v(\phi_t, \theta_t)\Delta t||_2^2}{4D^2\Delta t}, \tag{3.22}$$

and then simplifying we get

$$\frac{\frac{\partial L(\theta)}{\partial D}}{\ln\left(L(\theta)\right)} = \sum_{t=1}^{T} \frac{||\Delta X_t - v_{mag} v(\phi_t, \theta_t)\Delta t||_2^2 - 4D\Delta t}{4D^2\Delta t}, \tag{3.23}$$

or equivalently

$$D = \frac{\sum_{t=1}^{T} ||\Delta X_t - v_{mag} v(\phi_t, \theta_t)\Delta t||_2^2}{4T\Delta t}. \tag{3.24}$$

Similarly, we can differentiate equation 3.21 with respect to velocity magnitude and simplify to get

$$v_{mag} = \frac{\sum_{t=1}^{T} \Delta X_t v(\phi_t, \theta_t)}{\sum_{t=1}^{T} \Delta t ||v(\phi_t, \theta_t)||_2^2}. \tag{3.25}$$

Additionally, if there are more than one motion states present, which is the case with our model, there exists an indicator function, $\rho_{t,s}$ that is 1 if the particle is in state s at time i, and 0 otherwise, giving us the following equations for v_{mag} and D

$$D = \frac{\sum_{t=1}^{T} \rho_{t,s} ||\Delta X_t - v_{mag} v(\phi_t, \theta_t)\Delta t||_2^2}{\sum_{t=1}^{T} \rho_{t,s} 4\Delta t}, \tag{3.26}$$

$$v_{mag} = \frac{\sum_{t=1}^{T} \rho_{t,s} \Delta X_t v(\phi_t, \theta_t)}{\sum_{t=1}^{T} \rho_{t,s} \Delta t ||v(\phi_t, \theta_t)||_2^2}. \tag{3.27}$$

Given the complete data, we can construct a second indicator function which describes the transition that occurs at a given time. This function is given by $\tilde{\rho}_{t,\tilde{s}s}$ where $\tilde{\rho}_{t,\tilde{s}s} = 1$ if a the particle transitions from state s at time $t-1$ to state $\tilde{s}$ at time t and $= 0$ otherwise. From $\tilde{\rho}$ and ρ we can then calculate the matrix exponential for our transition rate matrix, Φ, discussed in Section 3.1.1, with the equation

$$\Phi_{\tilde{s},s} = \frac{\sum_{t=2}^{T} \tilde{\rho}_{t,\tilde{s}s}}{\sum_{t=2}^{T} \rho_{t-1,s}}. \tag{3.28}$$

From Φ we can calculate the transition rate matrix using a Taylor expansion in t of $\Phi = e^{tW}$.

These formulas for v_{mag}, D, and the transition rate matrix however require complete data, which in the case presented means that we need to know S_t allowing for the generation of the indicator functions. If this information is not known, additional work is needed. Thus, in models where we have multiple states and the state assignments are unknown at specific time points, a different approach is needed.

3.2.3 Models with Unobserved States

Stochastic processes in which state value is unknown have been called "doubly stochastic". This term refers to the fact that the model has two stochastic components, the first being the stochastic switches between states, and the second being the stochastic observations. Rabiner et al. in particular refer to doubly stochastic models as models with one stochastic process being unobserved and underlying. In the paper "An Introduction to Hidden Markov Models," Rabiner et al. define a Hidden Markov Model (HMM) to be one such doubly stochastic model. Beyond defining HMMs as doubly stochastic models, Rabiner only assigns three additional elements to this class of models. He states that HMMs must have a finite number of states, that there must be an observation at every time step, and that there is a state switch at each time step. State switches can however include a literal switch into a new state or remain in the previous state. This creates a concrete definition of a type of model without imposing too many restrictions on what specifically qualifies.

Within this framework, one can include models that are are time dependent as well as models with discrete observations or continuous observations.

Rabiner et al. solidify this definition with a classic example of an HMM, the coin toss problem. If someone is flipping one single fair coin, there are two observed states, heads and tails, which correspond to observations. There is one stochastic process and nothing is hidden. However, if we expand this problem to include an additional weighted coin, it gets more interesting. If someone were to flip one of the coins behind a barrier and only announce whether the coin lands on heads or tails and potentially switches coins between tosses, a doubly stochastic process is created. The states become the fair coin and the unfair coin, with a stochastic switching process happening between the two. The observations correspond to whether the stochastically determined heads or tails results occurs. Thus, a stochastic process dictates the observations as well.

Additionally, the coin flip problem meets each of Rabiner's three criteria, as we only have two states, an observation at each time point, and each time point corresponds to a state switch. Thus, this coin problem can be modeled with an HMM[26]. Because of the stochastic nature of the underlying equations, statistical tools can be utilized to answer some of the questions we have about the model. Namely, statistical methods allow us to estimate the parameters present in our model. In addition to being doubly stochastic, our model has a discrete number of states, discrete observations at every time step, and there is a possibility of state switching at every time step. Therefore, even though our process models are not inherently a HMM, we can use the statistical tools developed for HMMs on our models.

Rabiner et al. cites the computation of the probability of the observation sequence given the model, i.e. identifying the optimal state sequence, and optimizing the model parameters to be the main problems plaguing parameter estimation methods for models with latent states. Therefore, the computational expense is a problem we must consider. To address these problems, Rabiner et al. outline the steps and mechanisms of the Forward-Backward algorithm and Baum-Welch re-estimation formulas, two components of the Expectation-Maximization routine which we utilize for our models. These algorithms focus on calculating posterior distributions, such as the joint probability of the observations up until time T and the state at a given time t. The details of the Forward-Backward algorithm and Baum-Welch re-estimation will be explored in Section 3.2.4.

First, we need to expand on our understanding of Hidden Markov Models and their statistical relationship to our model by exploring Zoubin Ghahramani's 2001 paper "An Introduction to Hidden Markov Models and Bayesian Networks". The author defines a HMM as a tool for representing probability distributions over sequences of observations that has states which are hidden from the observer and satisfies the Markov property. This property allows for the decomposition of the conditional probability of the sequence of states and sequence of observations as follows:

$$P(S_{1:T}, X_{1:T}) = P(S_1)P(X_1|S_1)\prod_{t=2}^{T} P(S_t|S_{t-1})P(X_t|S_t), \tag{3.29}$$

where S_t is the state at time t and $X_{1:T}$ is the sequence of observations from time 1 to time T. The Markov property, as described in Section 3.1, thus

holds important implications in the statistical calculations we will explore later. Note that our model holds this property. The formal introduction of the Markov property adds the necessary specificity to complete the calculations needed for parameter estimations.

Just as Rabiner does, Ghahramani offers a strong look at Expectation Maximization through the lens of Forward-Backward and Baum-Welch algorithms. The author states that EM works by alternating between maximizing a lower bound of the likelihood function with respect to any distribution over the hidden variables and maximizing the parameters. Namely, during the E step we generate a distribution while the parameters are held constant and during the M step we maximize the parameters while the distribution is held constant. The author offers a derivation for EM by first expressing the log probability specifically as

$$\log(P(S_{1:T}, X_{1:T})) = \log(P(S_1)) + \sum_{t=1}^{T} \log(P(X_t|S_t)) + \sum_{t=2}^{T} \log(P(S_t|S_{t-1})). \tag{3.30}$$

We used this derivation as a foundational block for all the work we did with EM algorithm. In the next subsection, we will look at the various outputs and components generated by the algorithms we use.

3.2.4 Expectation Maximization

As MLE is not enough to estimate parameter values in cases with incomplete data, the Expectation Maximization (EM) algorithm can be used. The EM algorithm has 2 steps: the expectation step (E-step) and the maximization step (M-step). Both steps offer an important result. Specifically, the E-step

finds a distribution over the latent states while the M-step optimizes over the parameters. The M-step algorithm will look very similar to an MLE algorithm, with the distribution generated in the E-step integrated in. In the case of our model, the E-step produces the distribution of the probability of being in state S at time t. This distribution can take the place of the indicator functions which would be present in an MLE algorithm for our model with complete data. There are a number of algorithms for the E-step, but we will discuss one in particular, the Forward-Backward (FB) algorithm[11, 26].

It is important to note that the EM algorithm is an iteration. Initial guesses for the parameter values pass through the E-step to find probability distributions, which then pass through the M-step to update the parameter estimations. Only the distributions are updated in the E-step, but the E-step needs some parameter estimations to run. Similarly, the M-step only updates the parameter estimations, but it needs the distributions to run. The algorithm will continue alternating between the E-step and M-step, starting with the E-step, for some prescribed number of iterations or until some error bound is achieved on the M-step output.

By iterating over the E-step first and then the M-step, the EM algorithm works as a coordinate ascent method to find the maximum of the likelihood function[11, 6]. Additionally, because the E-step only adjusts the distribution and the M-step only adjusts the parameter values, we cannot decrease the likelihood after each combined EM iteration[11, 26, 6]. It can be rigorously shown that the likelihood is not only non-decreasing, but it is strictly increasing whenever we increase the distributions generated by the E-step[6]. This means that with each iteration of both E-step and M-step we approach a local

maximum for the set of parameters, and that at some point, we converge to that local maximum.

If we recall, MLE maximizes the likelihood function, given by

$$L(\theta) = \prod_t P(X_t|\theta).$$

However, in the case of incomplete data, the set of hidden variables, such as unobserved hidden states, S must also be considered, and we see

$$L(\theta) = \prod_t P(X_t, S_t|\theta). \tag{3.31}$$

Generalizing this to time sequence data for time $1:T$, we get the equation

$$\ln(L(\theta)) = \ln(\sum_{S_T} ... \sum_{S_1} \prod_{t=2}^{T} P(X_t, S_t|X_{t-1}, S_{t-1}, \theta) P(X_1, S_1|\theta)). \tag{3.32}$$

This formula is what our EM algorithm is optimizing through the iterative cycle.

The E-step consists of the forward-backward algorithm. The FB algorithm makes two complete passes over the data, once forward and once backward so that a complete view of the data is used in generating the probability distribution. Through these passes over the data, we calculate two probabilities, which we will call α and β. α and β will then be used to generate the probability distributions ultimately produced by the E-step[11, 26].

Let S_t be the state at time t and $X_{1:t}$ be the sequence of observations from X_1 to X_t. We can then represent the joint probability of S_t and $X_{1:t}$ as

$$\alpha_t = P(S_t, X_{1:t}), \tag{3.33}$$

which can be simplified into an iterative formula in the following way

$$\alpha_t = \sum_{S_{t-1}} P(S_{t-1}, X_{1:t-1}) P(S_t|S_{t-1}) P(X_t|S_t),$$

$$\alpha_t = \sum_{S_{t-1}} \alpha_{t-1} P(S_t|S_{t-1}) P(X_t|S_t), \tag{3.34}$$

where α_1 is some initial guess for the first state assignments. For example, α_1 can be taken to a uniform distributions across all states, or a distribution based on the initial guesses for the transition rates. We can then represent the conditional probability of $X_{t+1:T}$ given S_t as

$$\beta_t = P(X_{t+1:T}|S_t). \tag{3.35}$$

Similar to the α value, β can be simplified into the following iterative formula

$$\beta_t = \sum_{S_{t-1}} P(X_{t+2:T}|S_{t+1}) P(S_{t+1}|S_t) P(X_{t+1}|S_{t+1}),$$

$$\beta_t = \sum_{S_{t-1}} \beta_{t+1} P(S_{t+1}|S_t) P(X_{t+1}|S_{t+1}), \tag{3.36}$$

where β_T is initialized to be a vector such that every entry, $b_t = 1$.

The calculation of α and β completes the Forward-Backward algorithm, but we are not yet ready to go into the M-step[11, 26]. As mentioned earlier, the E-step has one more component, taking our α and β values over time and using them to generate the two expectations needed to complete the M-step.

The first expectation, $P(S_t|X_{1:T},\theta)$, can be seen as the probability of being in state S at time t given the parameters and sequence of observations. At each time point t, it is a vector with the number of entries equal to the number of states[11, 26, 6].

$$P(S_t|X_{1:T},\theta) = \frac{P(S_t, X_{1:T})}{\sum_{S_t} P(S_t, X_{1:T})} = \frac{P(S_t, X_{1:t})P(X_{t+1:T}|S_t)}{\sum_{S_t} P(S_t, X_{1:t})P(X_{t+1:T}|S_t)} = \frac{\alpha_t\beta_t}{\sum_{S_t}\alpha_t\beta_t}. \tag{3.37}$$

Similarly, the second expectation, $P(S_t, S_{t+1}|X_{1:T},\theta)$, is the joint probability of being in state i at time t and j at time $t-1$ and the sequence of observations. At each time point t, it is a n by n matrix where n is the number of states.

$$\begin{aligned} P(S_t, S_{t+1}|X_{1:T},\theta) &= \frac{P(X_{1:T}, S_t, S_{t-1})}{\sum_{S_t}\sum_{S_{t-1}} P(X_{1:T}, S_t, S_{t-1})} \\ = \frac{P(S_{t-1}, X_{1:t-1})P(S_t|S_{t-1})P(X_t|S_t)P(X_t|S_t)}{\sum_{S_t}\sum_{S_{t-1}} P(S_{t-1}, X_{1:t-1})P(S_t|S_{t-1})P(X_t|S_t)P(X_t|S_t)} &= \frac{\alpha_{t-1}\Phi_{t,t-1}P(X_t|S_t)\beta_t}{\sum_{S_t}\sum_{S_{t-1}}\alpha_{t-1}\Phi_{t,t-1}P(X_t|S_t)\beta_t}, \end{aligned} \tag{3.38}$$

where Φ is the matrix exponential as defined in Section 3.5.

From here, we can combine the concepts of the MLE formulas given in equation 3.26 and equation 3.27 and this expectation $P(S_t|X_{1:T},\theta)$ to calculate the parameters for each state in our model[11, 26, 6]. Namely, where we have our indicator function, $\rho_{t,s}$, we will now have $P(S_t|X_{1:T},\theta)$. As this expectation represents the probability of being in state S at time t, it serves here as a quasi-indicator function or a weight on the observations at that time step. For instance, the probability that a set of observations at a given time point, lets

call it τ, is a result of a cell being in a given diffusion state, s_D, is given by $P(S_\tau = s_D|X_{1:T}, \theta)$. This would then be reflected in our calculation of D, as these observations would be expressed proportionally to that probability. Making this adjustment to the MLE formulas given in equation 3.26 produces the following equation

$$D = \frac{\sum_{t=1}^{T} P(S_t = s_D|X_{1:T}, \theta)||\Delta X_t||_2^2}{\sum_{t=1}^{T} P(S_t = s_D|X_{1:T}, \theta)4\Delta t}. \tag{3.39}$$

Similarly, given $\mathbb{W}$ is the set of states corresponding to a swim state (after discretizing the continuous direction variables ϕ and θ) so that a single swim state is comprised of multiple, discrete states as described in Section 3.1.4, equation 3.27 can be adjusted to get

$$v_{mag} = \frac{\sum_{s\in\mathbb{W}}\sum_{t=1}^{T} P(S_t = s|X_{1:T}, \theta)v(\phi_t, \theta_t)\Delta X_t}{\sum_{s\in\mathbb{W}}\sum_{t=1}^{T} P(S_t = s|X_{1:T}, \theta)\Delta t||v(\phi_t, \theta_t)||_2^2}, \tag{3.40}$$

$$D_2 = \frac{\sum_{s\in\mathbb{W}}\sum_{t=1}^{T} P(S_t = s|X_{1:T}, \theta)||\Delta X_t - v_{mag}v(\phi_t, \theta_t)\Delta t||_2^2}{\sum_{s\in\mathbb{W}}\sum_{t=1}^{T} P(S_t = s|X_{1:T}, \theta)4\Delta t}, \tag{3.41}$$

where s in v_{mag} and D_2 are the swim states. Additionally, we can take equation 3.28 and replace the indicator function $\tilde{\rho}$ from Section 3.2.2 with $P(S_t, S_{t+1}|X_{1:T}, \theta)$ to get the updated equation for the matrix exponential Φ. Given states $\tilde{s}$ and s, we see that

$$\Phi_{\tilde{s},s} = \frac{\sum_{t=2}^{T} P(S_{t-1} = s, S_t = \tilde{s}|X_{1:T}, \theta)}{\sum_{t=2}^{T} P(S_{t-1} = s|X_{1:T}, \theta)}. \tag{3.42}$$

By combining the MLE formulas with the expectations generated by the FB algorithm, EM pulls from the established concept of MLE with a complete

view of incomplete data. Since $P(S_t|X_{1:T}, \theta)$ is calculated with α_t and β_t, it creates an accurate view of the data in relation to the potential states. This is because α_t and β_t view the data set as a whole and pass through the observations completely forwards and backwards, respectively. Notice that all of the MLE results given by equations 3.26 and 3.27 weight each track by the number of increments. Hence, EM naturally results in by-increment estimations.

The use of EM reveals important information about the *Salmonella*. Not only does the M-step estimate the diffusion values, velocity magnitude, and transition rates between states, but the E-step also estimates important information. In particular, the E-step produces the distributions $P(S_t|X_{1:T}, \theta)$ and $P(S_t, S_{t+1}|X_{1:T}, \theta)$, given by equations 3.37 and 3.38 which give valuable information about how the cells are behaving. From $P(S_t|X_{1:T}, \theta)$, we can discover the most probable state a cell is in at a given time and from $P(S_t, S_{t+1}|X_{1:T}, \theta)$ we can discover the most probable transition at a given time[11, 26]. From this information, we can observe the tracks as states through time, as well as overlay the states into the video. This visualization of the model results allows us to explore how well each model describes the motion seen in the video data. To validate our EM code, we use it to process simulated data, in order to test the results at each step. By utilizing the Gillespie algorithm to simulate the stochastic state switches, we are able to create tracks which accurately mimicked the experimental data we have, and since we know the expected parameter values, we could confirm the accuracy of our EM code.

3.2.5 Information Criteria

When considering multiple models in the context of a give dataset, one can use a so-called information criterion (IC). On the surface, information criterion seem to hold similar goals to ours. An IC assigns a real-valued score to a model based on certain criterion such as model complexity, parameter fit, and other measurable factors. Popular IC include but are not limited to Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). These IC are designed specifically to compare models by providing a cost function that balances model complexity with predictive accuracy. The idea is to find a model with the simplest structure and smallest number of parameters that also provides a good match the data. Models with fewer parameters avoid the problem of overfitting and are typically better able to describe new data. For this reason, models that have optimal IC scores are sometimes thought to be more reflective of the truth. However, taking without imagination their purpose as dictated by their theoretical formulations, IC are merely sophisticated measures of predictive power. In statistics predictive accuracy is formulated in a concrete way, for example using the Kullback-Leibler distance, and does not necessarily consistently yield higher scores for models that are consistent with causality and/or known physical laws. Rather than relying on IC to compare models, we seek to perform a qualitative model comparison that synthesizes a range of factors, including our knowledge of how Salmonella are known to operate, with the goal of better understanding motion heterogeneity.

3.3 Validation of Statistical Tools Using Generative Simulations

3.3.1 An Adjustment to the Gillespie Algorithm

While the Gillespie algorithm is fundamental to simulating data for our model, it cannot give us everything we need. The algorithm can be used to simulate which state a cell is in at time t, as well as the times at which switches happen, but it cannot simulate the change in position experienced by the particle. To do this, we must first consider a fixed time step, $\Delta t > 0$, so that the position of our simulated particle can be recorded at every time step. While this step size does not need to be constant, it is helpful for the step size to be small, so that a few observations can be observed prior to a switch, however this is not necessary. Then, for every time step, a normal random variable with mean 0 and variance Δt should be pulled. The track can then be propagated forward utilizing the SDE for the given state as well as the random variable pulled. For example, if the particle is in the dormant state, for each time point $X(t) = X(t-\Delta t)+\sqrt{2D_0}N(0,\Delta t)$. SDEs for all states in the three-state model can be seen in equation 3.14.

This forward propagation of the particles position is continued until time $t+\tau$ where $t+\tau$ is the time of a switch to a new state. At that point, the SDE used to calculate position is updated. This continues for a certain number of time points or a specific length of time. A sample trajectory can be seen in Figure 3.2. The simulation of data and testing of our EM algorithm on the simulated data is an important step in this project as it allowed us to confirm

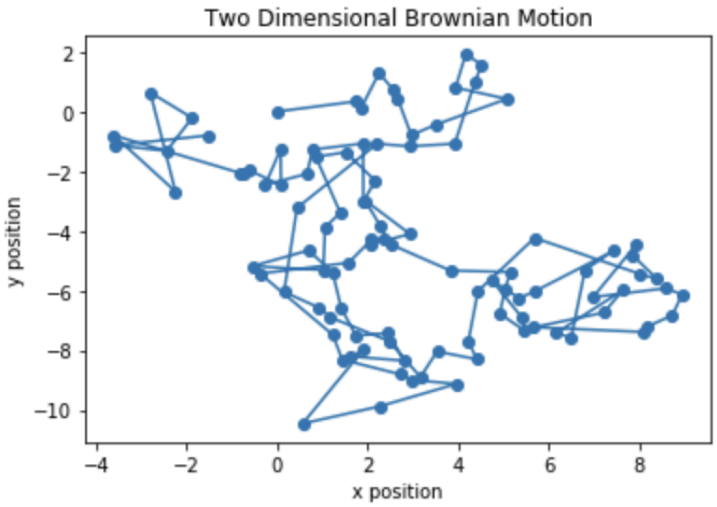

Figure 3.2: A Sample Brownian Motion Trajectory in 2D

the accuracy of our algorithm prior to applying it to experimental data.

3.3.2 Building up to Full Model by Testing EM and MLE with Simulated Data

Because there is no way to test the accuracy of the EM algorithm directly with the wild type data as the true state and parameter values are unknown, creating simulated data played an important role in creating and extending the model. All simulated data is created with the modified Gillespie algorithm explained above. By starting out with a single simulated track of a particle experiencing a diffusion-only driven random walk, we are able to validate our Expectation Maximization algorithm. Adding the complexity layers individually and testing the accuracy of the parameter estimation along the way played a vital role in model development.

To start this process, the diffusion coefficient of a single particle experienc-

ing diffusion-driven random walk in two dimensions is calculated via Maximum Likelihood Estimation. After that, we adjust the particle to experience switching diffusion. These switches happen at random time points, facilitated by the Gillespie Algorithm. As this simulation proceeds, we record not only the position data at every time point but also which diffusion coefficient the particle had at every time point. With this information we are again able to use MLE to calculate both diffusion coefficients. Additionally, we are able to start the process of developing our EM algorithm. Given that we know all of the parameter values and the diffusion value at the start of the simulation, the EM algorithm can be checked at every step.

From here, we add a third state the particle could switch into. This state is a swim state in two dimensions. The EM algorithm is subsequently updated to include the estimation for velocity magnitude and the corresponding diffusion value. Once that version of the EM algorithm is rigorously tested, we add a third dimension to our simulations. Again, our EM algorithm is tested with this update to the simulations. The next update came only to the EM algorithm and not the simulations however. Given the nature of the data we have, i.e. the tracking data from microscopy data as described in Section 2.5, we know we will be dealing with cells which have three dimensional movement but only two dimensional data. In order to account for this fact, we simulate 3D tracks and fit an EM algorithm which only uses 2D data. To do this, we adjust the M-step calculations of velocity magnitude by dividing by $||\tilde{v}(\phi_t, \theta_t)||_2^2$ where $\tilde{v}(\phi_t, \theta_t)$ is the x and y components of $v(\phi_t, \theta_t)$. This changes the formula

given by equation 3.40 to

$$v_{mag} = \frac{\sum_{s\in\mathbb{W}}\sum_{t=1}^{T} P(S_t = s|X_{1:T},\theta)\tilde{v}(\phi_t,\theta_t)\Delta X_t}{\sum_{s\in\mathbb{W}}\sum_{t=1}^{T} P(S_t = sX_{1:T},\theta)\Delta t(\sin(\theta_t))^2}, \tag{3.43}$$

where θ is the vertical angle on the unit sphere corresponding to a given discretized state s.

To confirm our algorithm works for the amount of data we have, additional simulated tracks are created and subsequently ran through the EM algorithm. Additionally, any time we extended the model to try to capture population heterogeneity, the EM algorithm is always first tested and debugged on simulated data in this manner. In particular, for the heterogeneous movement pattern models we present in Section 4.2, tracks are simulated as being from one population or the other, i.e. none of the data had switches between population. However, the structure of our EM algorithm accounts for the possibility of switching between tracks, and we confirm that the EM algorithm would predict the populations as phenotypically separate by setting the initial guesses for these rates to be non zero. Therefore, as we iterate through our EM algorithm, we can test to make sure the algorithm predicts the populations as disjoint by confirming the estimations for those rates approach zero with each iteration. By testing the parameter estimation tool on simulated data in this way, we are able to gain confidence in the fit and results produced on experimental data.

To demonstrate the accuracy of our EM algorithm on simulated data, Figure 3.3, Figure 3.4, and Figure 3.5 were created. For this graphic, we use the four-state model discussed in Section 4.2 and 3D movement restricted to

2D data. We simulated one track from each of the two populations for a combined total of 10,000 increments. The increments were divided according to the indicated fraction in the bottom row of Figure 3.5. For example, the tracks that correspond to 30% diffusers and 70% swimmers have 3,000 and 7,000 increments, respectively.

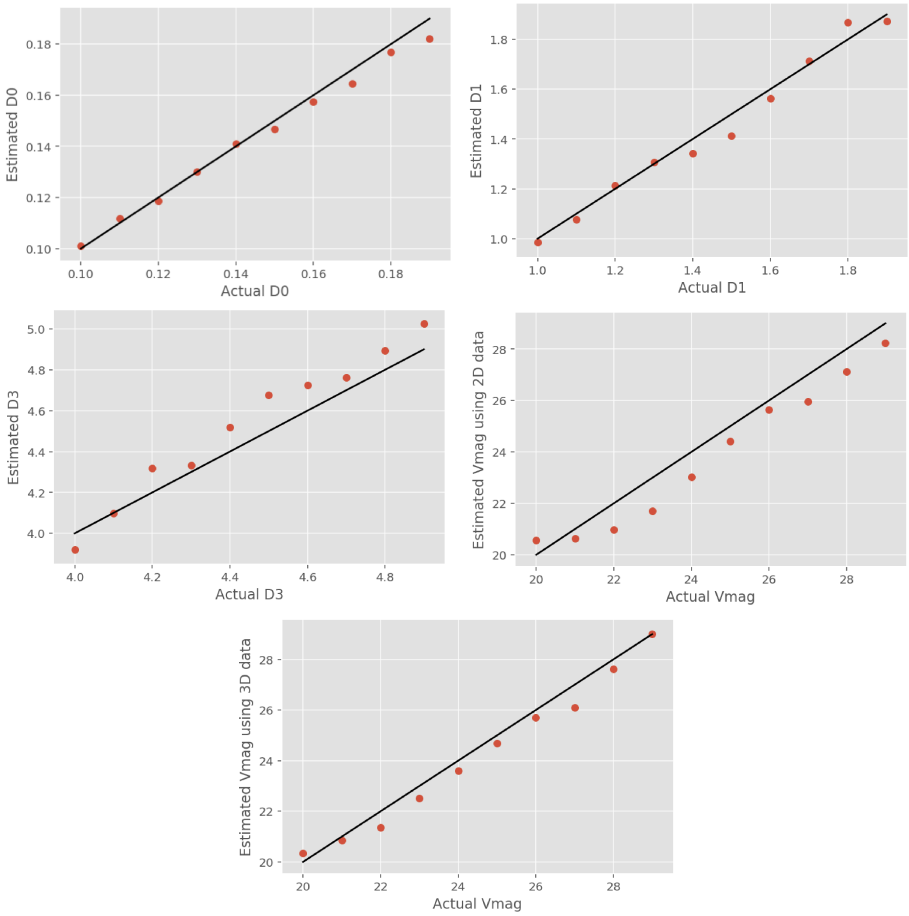

Figure 3.3: Accuracy of the Expectation Maximization algorithm is demonstrated by low error on the estimation of parameters with the four-state model. The scatter plot points are the actual value plotted against the estimated values, and the line is the $x = y$ line to demonstrate how close to truth the estimates are. Top left is D_0, top right is D_1, row 2 left is D_3, and row 2 right is v_{mag} in 3D with 2D data. The bottom is v_{mag} with 3D data. The simulations for these estimations used 10,000 time points.

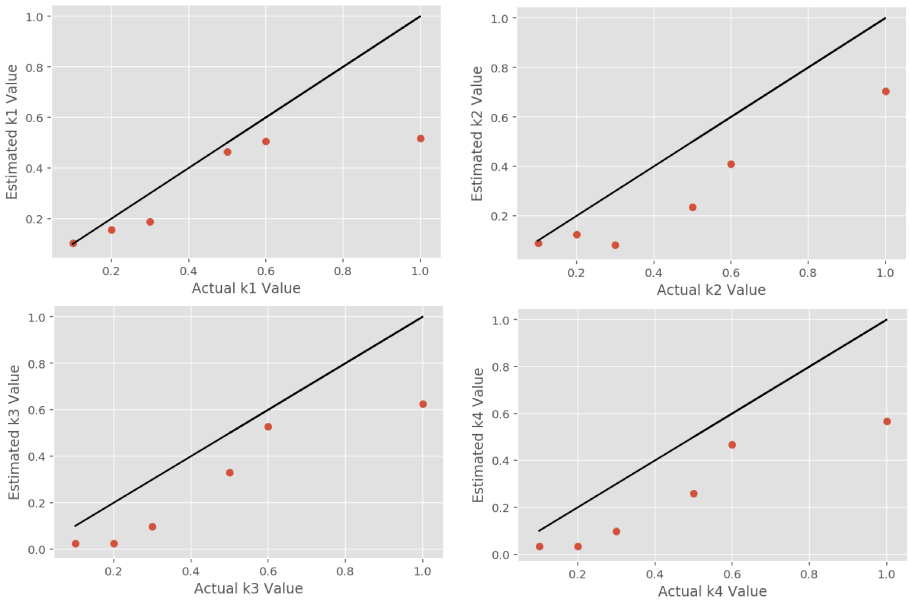

Figure 3.4: Accuracy of the Expectation Maximization algorithm is demonstrated by low error on the estimation of parameters with the four-state model. For the first two rows, the scatter plot points are the actual value plotted against the estimated values, and the line is the $x = y$ line to demonstrate how close to truth the estimates are. Top left is k_1, top right is k_2, row 2 left is k_3, and row 2 right is k_4.

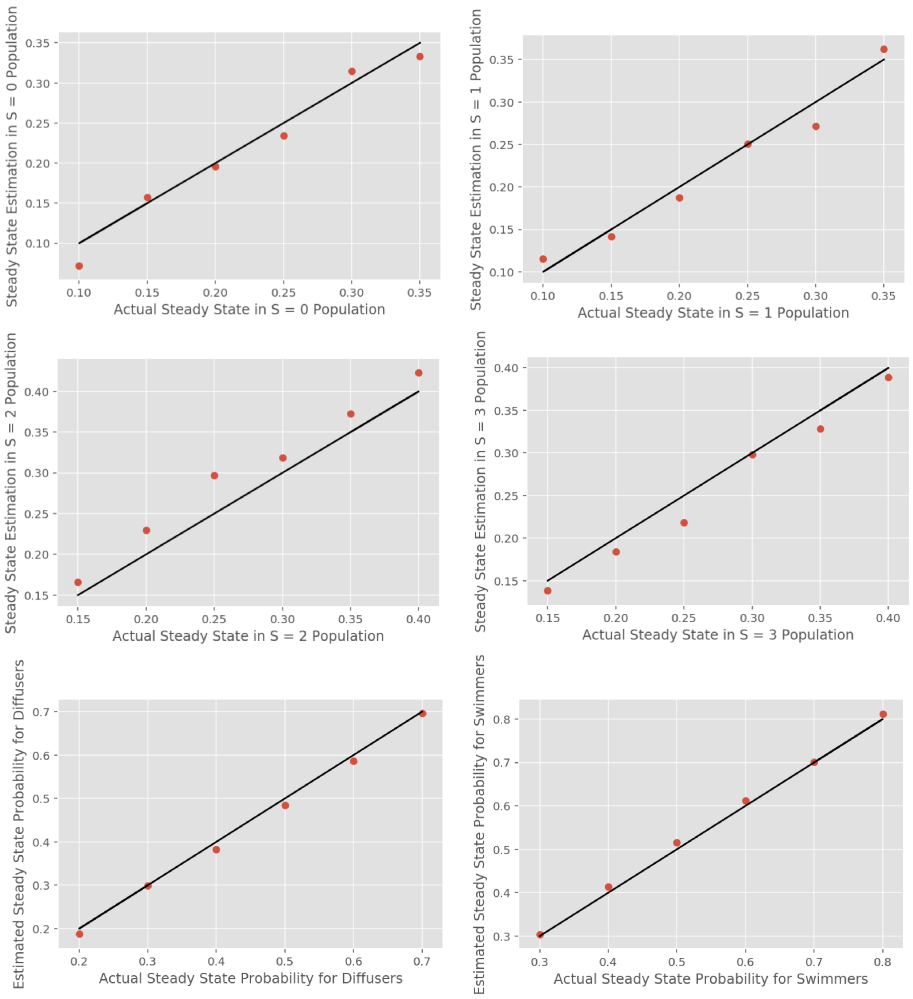

Figure 3.5: Accuracy of the Expectation Maximization algorithm is demonstrated by low error on the estimation of parameters with the four-state model. For the first two rows, the scatter plot points are the actual value plotted against the estimated values, and the line is the $x = y$ line to demonstrate how close to truth the estimates are. Top left is the steady state probability for $S = 0$, top right is the steady state probability for $S = 1$, row 2 left is the steady state probability for $S = 2$, and row 2 right is the steady state probability for $S = 3$. The bottom is the fraction that are actually in the diffusers (left) and swimmers (right) populations plotted against the steady state estimation for the diffusers and swimmers populations.

Chapter 4

Results: Stochastic Models for Characterizing Heterogeneity of *Salmonella* Motion

In this chapter we will explore the results of the three-state model, given by equations 3.13 and 3.14, on wild type cell data. After discussing the shortcomings of these results we will outline and explore the various heterogeneous models we develop. Finally, we will discuss the biological implications of the parameter estimations as well as visualize the E-step results by examining the most probable state at a given time t.

Parameter	Estimated Value
D_0	0.0143 $\mu m^2/sec$
D_1	0.1357 $\mu m^2/sec$
D_2	1.1981 $\mu m^2/sec$
v_{mag}	19.743 $\mu m/sec$
k_1	0.6339 $transitions/sec$
k_2	0.4117 $transitions/sec$
k_3	0.4125 $transitions/sec$
k_4	1.5967 $transitions/sec$
$P(S_t = 0)$	0.34
$P(S_t = 1)$	0.525
$P(S_t = 2)$	0.135

Table 4.1: Parameter Estimations for the three-state Model, given by equations 3.13 and 3.14 (see Figure 3.1)

4.1 Results of Applying Expectation Maximization to the Homogeneous Population Model and Wild Type Cell Data

The three-state model, given by equations 3.13 and 3.14, is applied to the wild type data set in a few different ways. First, the algorithm is applied to the individual duodenum, ileum, and jejunum data sets for each mouse. After, the data is grouped together by GI tract section over all mice. We then consider the data set as a whole, combining the data from all mice and all sections of the GI tract together. The results for the combined view of the data can be found in Table 4.1

The results offer both insights into the nature of heterogeneity in the population. While a large portion of the resulting parameter estimations are relatively similar, there are a few noticeable outlying mice whose data for specific

GI track sections do not follow the general trend. With the velocity estimations for cells in specific mice ranging from $0.1049\mu m/sec$ to $26.83\mu m/sec$, it is clear that while some cells run and tumble as expected, other groups of cells experience almost exclusively undirected motion. While some of the anomalous parameter estimations can be explained by a small number of data points, other anomalous mice appear to have sufficient data points. Thus, the microscopy videos for these mice are reexamined. Overwhelmingly, these outlier mice appear to have less directed motion *Salmonella* than the rest of the samples, which supports the lower velocity magnitude estimations for the cells in these mice.

Interestingly, even though the velocities differ, all of the transition rate estimations are similar. This is unexpected due to the nature of Markov processes. Since Markov processes are "memory-less," we should see cells transitioning through all three-states in a single track given the estimated transition rates (see 4.1). However, we observe that most cells either switch exclusively between dormant and tumble or exclusively between tumble and swim, which violates the Markov property, as it implies memory. That is, if a cell has swam in the past, it appears more likely to swim in the future, breaking the Markov property.

We also examine the steady state probabilities for our stochastic process by solving the linear system $0 = WP_\infty$ given by equation 3.10 in Section 3.1.2 for P_∞ and substituting our transition rate matrix from equation 3.13 for W. To do this, we used Cramer's Rule to solve the linear system. This investigation reveals that about 34% of cells are expected to be in the dormant state at any given time, with 52% of cells in the tumble state and the remainder in

the swim state. This prediction matches a by-increment analysis, where the most probable state is examined at each time. We expect the model to assign cells to the dormant state whenever an extended period of undirected motion occurs, as that state is designed for the cells with the lowest diffusivity and most undirected motion, but the steady states suggest a large portion of those cells are assigned to tumble. These observations offer insight for the nature of a phenotypic heterogeneity which can be observed in motion patterns within our experimental data. In an attempt to see if this trend is observable in the video data, all of the videos in the wild type set are then reexamined through a heterogeneous population lens. It is found that cells appear to be predominantly dormant or predominantly swimming in the videos, or that two sub-populations are present.

We also visualize the data as tracks through time in an attempt to examine how different the potential populations are. Utilizing ImageTank, a program specifically designed to view and visualize scientific images, and the distributions generated by the E-step of the EM algorithm, we are able to create useful visualizations[35]. These figures, seen in Figure 4.1, illustrate the stark difference between cells with directed motion and cells with undirected motion. These visual results support our previous conclusions from inconsistent parameter estimations across individual mice as well as the steady state predictions placing so many cells in the tumble state. The observation of two motion patterns leads us to start thinking about our data as two heterogeneous sub-populations. This in turn leads to the exploration of model extensions that explicitly model multiple sub-populations.

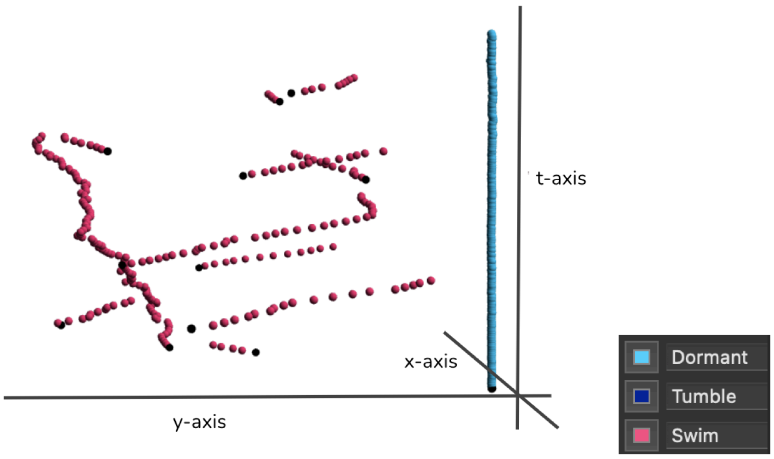

Figure 4.1: Left: An example of the state assignments from the three-state model, given by equations 3.13 and 3.14, Right: Legend.

4.2 Heterogeneous Population Model

4.2.1 Development of a Heterogeneous Population Model

Given the evidence for two sub-populations, we decide to extend the original three-state model, given by equations 3.13 and 3.14, to represent two groups with three states each. We extend our homogeneous population model into a heterogeneous population model by adding three additional states. The model is considered a heterogeneous model because each sub-populations represent a part of the population which is phenotypically heterogeneous from the other sub-population. By doing this, we create a model based on the homogeneous model that has a dormant, tumble, and swim state for two sub-populations. The idea being that each population still has similar motion types to what we predict, with slightly different diffusivity and velocity values. The model structure can be observed in Figure 4.2 and the SDEs for this model follow the same structure as the three-state model, given by equations 3.13 and 3.14.

Similar to the homogeneous model, there would be transitions between dormant and tumble as well as tumble and swim within each population. However, there would also be the additional possibility of transition between populations, either dormant to dormant or tumble to tumble. Through the iterations of EM, if these states represent sub-populations in the way we expect, the inter-population transition rates should converge to near zero values as the M-step maximizes the parameter values, suggesting no transitions between the two populations. That is, we set up an irreducible transition rate matrix that should converge in the coordinate ascent sense described in Section 3.2.4 to a near reducible transition rate matrix (or a matrix which is similar to a block upper triangular matrix when simplified) when applied to the data via EM algorithm. This model has master equation

$$\frac{\partial P(t)}{\partial t} = \begin{bmatrix} -k_1 - k_9 & k_2 & 0 & k_{10} & 0 & 0 \\ k_1 & -k_2 - k_3 - k_{11} & k_4 & 0 & k_{12} & 0 \\ 0 & k_3 & -k_4 & 0 & 0 & 0 \\ k_9 & 0 & 0 & -k_5 - k_{10} & k_6 & 0 \\ 0 & k_{11} & 0 & k_5 & -k_6 - k_7 - k_{12} & k_8 \\ 0 & 0 & 0 & 0 & k_7 & -k_8 \end{bmatrix} P(t) \tag{4.1}$$

and the following SDEs for each state

$$dX = \begin{cases} \sqrt{2D_0}dW, & S(t) = 0 \\ \sqrt{2D_1}dW, & S(t) = 1 \\ v_2(\phi_t, \theta_t)dt + \sqrt{2D_2}dW, & S(t) = 2 \\ \sqrt{2D_3}dW, & S(t) = 3 \\ \sqrt{2D_4}dW, & S(t) = 4 \\ v_5(\phi_t, \theta_t)dt + \sqrt{2D_5}dW, & S(t) = 5. \end{cases} \tag{4.2}$$

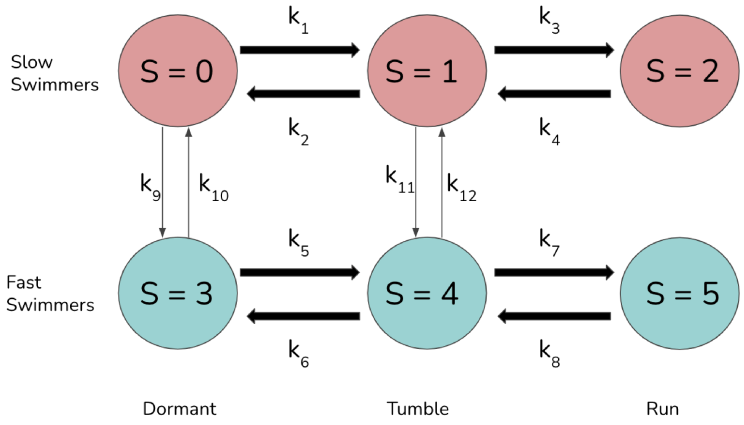

Figure 4.2: Two Swimming Populations Model Diagram

After running the wild type data through the EM algorithm updated for this two swimming populations model, given by equations4.1 and 4.2, we find some interesting results, which can be seen in Table 4.2. With this model, two of the parameter estimation values suggest this model may not reflect the biology. The velocity magnitude estimation for the slow swimmer population, that is the velocity magnitude estimation for state 2, is estimated to be a small

value, with the velocity estimating to around $0.7988 \mu m/sec$. This is an almost negligibly small value for the velocity magnitude to take especially if compared to state 5, which has a velocity magnitude of $22.7064 \mu m/sec$. Additionally, the rate of switching to the fast tumble, state 4, from slow tumble, state 1, is higher than the other inter-population transitions. This rate, k_{11} in Figure 4.2, is high enough at around 0.43 $transitions/sec$ to be comparable to the intra-population transition rates, indicating over time more cells will transition into the fast population. This contradicts the prediction of heterogeneous populations, as well as contradicts the video data which suggests the majority of cells are divided between the slow and fast populations.

Given the inconsistencies which arise from the two swimming population model, given by equations 4.1 and 4.2, we propose a change to the slow swim state, state 2. Since the velocity magnitude of this state is so low, we adjust it to be a second diffusion state, with a faster diffusion rate than the tumble state for that population. The updated model diagram is in Figure 4.3, and this model will be referred to as the diffusers and swimmers model. We hope this would lower the k_{11} rate, as it gives state 1 another, faster diffusion state to transition into, acting as an alternative to transitioning into state 4. Giving

Parameter	Estimated Value
D_0	0.001256 $\mu m^2/sec$
D_1	0.007389 $\mu m^2/sec$
D_2	0.06949 $\mu m^2/sec$
v_2	0.7988 $\mu m/sec$
D_3	0.02658 $\mu m^2/sec$
D_4	0.2231 $\mu m^2/sec$
D_5	1.1044 $\mu m^2/sec$
v_5	22.7064 $\mu m/sec$
k_1	2.141 $transitions/sec$
k_2	0.60902 $transitions/sec$
k_3	0.1165 $transitions/sec$
k_4	0.06656 $transitions/sec$
k_5	0.6028 $transitions/sec$
k_6	0.40804 $transitions/sec$
k_7	0.86701 $transitions/sec$
k_8	14.2730 $transitions/sec$
k_9	0.03748 $transitions/sec$
k_{10}	0.009943 $transitions/sec$
k_{11}	0.43 $transitions/sec$
k_{12}	0.04785 $transitions/sec$
$P(S_t = 0)$	0.030
$P(S_t = 1)$	0.1043
$P(S_t = 2)$	0.1826
$P(S_t = 3)$	0.2662
$P(S_t = 4)$	0.3933
$P(S_t = 5)$	0.02389

Table 4.2: Parameter Estimations for the Two-Swimming Populations Model from Figure 4.2.

the same master equation

$$\frac{\partial P(t)}{\partial t} = \begin{bmatrix} -k_1 - k_9 & k_2 & 0 & k_{10} & 0 & 0 \\ k_1 & -k_2 - k_3 - k_{11} & k_4 & 0 & k_{12} & 0 \\ 0 & k_3 & -k_4 & 0 & 0 & 0 \\ k_9 & 0 & 0 & -k_5 - k_{10} & k_6 & 0 \\ 0 & k_{11} & 0 & k_5 & -k_6 - k_7 - k_{12} & k_8 \\ 0 & 0 & 0 & 0 & k_7 & -k_8 \end{bmatrix} P(t) \tag{4.3}$$

but the following SDEs for each state

$$dX = \begin{cases} \sqrt{2D_0}dW, & S(t) = 0 \\ \sqrt{2D_1}dW, & S(t) = 1 \\ \sqrt{2D_2}dW, & S(t) = 2 \\ \sqrt{2D_3}dW, & S(t) = 3 \\ \sqrt{2D_4}dW, & S(t) = 4 \\ v_5(\phi_t, \theta_t)dt + \sqrt{2D_5}dW, & S(t) = 5. \end{cases} \tag{4.4}$$

This variation of a six-state model also leads to parameter estimations, which can be viewed in Table 4.3, that appear inconsistent with the video data. Namely, the transition rates appear inconsistent with what we would expect of a stochastic process model representing the video data. To start, the transition rates between swimming dormant, state 3, and tumble, state 4, rates k_5 and k_6 in Figure 4.3 are smaller than any rate estimation we have seen

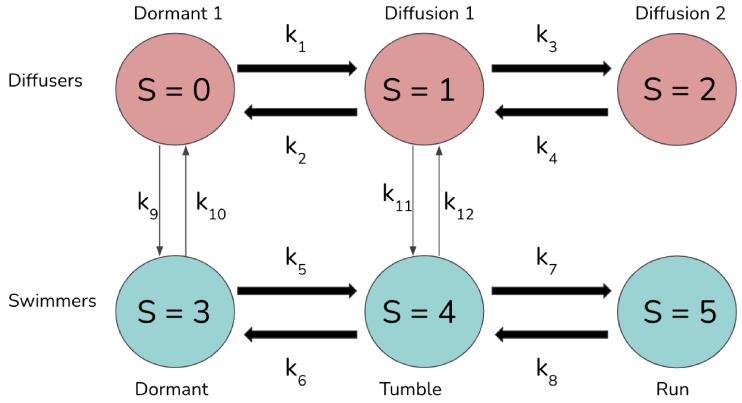

Figure 4.3: Diffusers and Swimmers Model Diagram

up until this point, around $3.0073e^{-05}$ transitions/sec. This suggests that very few cells transition between the dormant and the tumble states in the swimmer population. Additionally, all transition rates between the populations, $k_9 - k_{12}$, are higher than we expect. In this case, those four rates are comparable to the intra-population transition rates, suggesting that the cells switched between populations as much as within populations. This would indicate that there is not multiple sub-populations present, directly contradicting the visual data present in the videos.

Next, we decide to focus on addressing the issue of the low k_5 and k_6 rates. These rates are so low that there are little to no switches between state 3 and state 4 in the diffusers and swimmers model, given by equations 4.3 and 4.4. By combining this clue with the transition rate estimations for the three-state model, given by equations 3.13 and 3.14, and our observations about the video data, we decide to update our model structure to cut out the dormant state from the swimmer population. Additionally, the only way a cell could transition between populations is through the diffusion 1 state of the diffuser

Parameter	Estimated Value
D_0	0.00149 $\mu m^2/sec$
D_1	0.01716 $\mu m^2/sec$
D_2	0.06749 $\mu m^2/sec$
D_3	0.0072017 $\mu m^2/sec$
D_4	0.2275 $\mu m^2/sec$
D_5	1.0563 $\mu m^2/sec$
v_{mag}	24.3719 $\mu m/sec$
k_1	0.0607012 $transitions/sec$
k_2	0.009698 $transitions/sec$
k_3	0.9656 $transitions/sec$
k_4	0.6784 $transitions/sec$
k_5	0.00013914 $transitions/sec$
k_6	.0000030073 $transitions/sec$
k_7	1.436 $transitions/sec$
k_8	14.36 $transitions/sec$
k_9	0.2038 $transitions/sec$
k_{10}	0.1159 $transitions/sec$
k_{11}	1.556 $transitions/sec$
k_{12}	0.6544 $transitions/sec$
$P(S_t = 0)$	0.032
$P(S_t = 1)$	0.202
$P(S_t = 2)$	0.289
$P(S_t = 3)$	0.0763
$P(S_t = 4)$	0.341
$P(S_t = 5)$	0.059

Table 4.3: Parameter Estimations for the Diffusers and Swimmers Model (see Figure 4.3)

population and the tumble state of the swimmer population. The new model, referred to as the five-state model, is depicted in Figure 4.4. The idea with this update is to capture more fully the idea that cells either experienced predominantly directed motion or predominantly undirected motion. This change is reflected in our master equation

$$\frac{\partial P(t)}{\partial t} = \begin{bmatrix} -k_1 & k_2 & 0 & 0 & 0 \\ k_1 & -k_2 - k_3 - k_7 & k_4 & k_8 & 0 \\ 0 & k_3 & -k_4 & 0 & 0 \\ 0 & k_7 & 0 & -k_5 - k_8 & k_6 \\ 0 & 0 & 0 & k_5 & -k_6 \end{bmatrix} P(t) \quad (4.5)$$

and the following SDEs for each state

$$dX = \begin{cases} \sqrt{2D_0}dW, & S(t) = 0 \\ \sqrt{2D_1}dW, & S(t) = 1 \\ \sqrt{2D_2}dW, & S(t) = 2 \\ \sqrt{2D_3}dW, & S(t) = 3 \\ v_4(\phi_t, \theta_t)dt + \sqrt{2D_4}dW, & S(t) = 4. \end{cases} \quad (4.6)$$

At first glance, this five-state model, given by equations 4.5 and 4.6, seems to do a good job of classifying the motion and fitting parameters to the data. The parameter estimations can be viewed in Table 4.4. However, we still have some slight inconsistencies in the transition rates. k_8, the rate of switching from tumble to diffusion 1 in Figure 4.4 is comparable to k_4, the transition

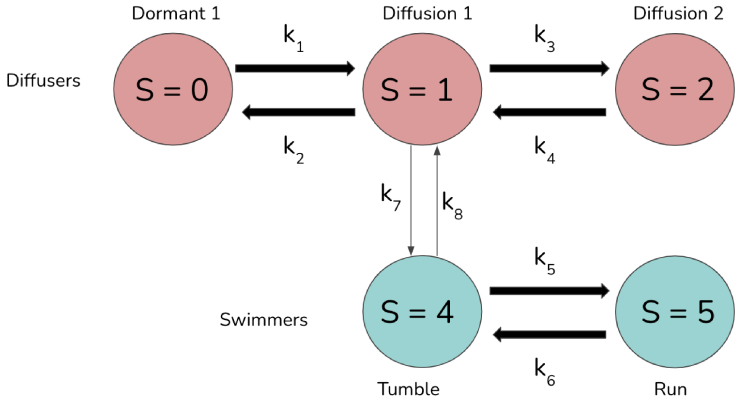

Figure 4.4: Five-State Diffusers and Swimmers Population Model

from diffusion 2 to diffusion 1. This is interesting to us because in previous versions of the heterogeneous population model, it is the transition rate into the faster population that is higher, not the transition rate out of it. This means that in both of the six-state models, the steady state estimations predict more cells in the swimming population while the five-state model predicts more cells in the diffusers population. Upon further investigation, we also find that the transition rate estimations between the populations are similar to the transition rates between diffusion 1 and diffusion 2. This suggests to us that cutting the second diffusion state from the diffusers population could strengthen the model. This change reduces the possibility for uncertainty since the transition rate estimations imply that the five-state model is sending cells which should be in a fast diffusion only state into the tumble state or visa versa.

With this, the fourth heterogeneous population model, aptly named the four-state model, is born. Depicted in Figure 4.5, this model still consists of two populations which can be referred to as diffusers and swimmers. The

Parameter	Estimated Value
D_0	0.034 $\mu m^2/sec$
D_1	0.2486 $\mu m^2/sec$
D_2	0.8996 $\mu m^2/sec$
D_3	0.385 $\mu m^2/sec$
D_4	1.529 $\mu m^2/sec$
v_{mag}	27.385 $\mu m/sec$
k_1	0.6758 $transitions/sec$
k_2	1.478 $transitions/sec$
k_3	0.0679 $transitions/sec$
k_4	0.5179 $transitions/sec$
k_5	2.9905 $transitions/sec$
k_6	0.9441 $transitions/sec$
k_7	0.0437 $transitions/sec$
k_8	0.472 $transitions/sec$
$P(S_t = 0)$	0.59
$P(S_t = 1)$	0.27
$P(S_t = 2)$	0.035
$P(S_t = 3)$	0.025
$P(S_t = 4)$	0.08

Table 4.4: Parameter Estimations for the five-state Model (see Figure 4.4)

diffusers population has a "diffusion 1" state and a "diffusion 2" state while the swimmer population has a tumble and a swim state. The goal with this model is to combine what we learned from the previous models with what we see in the video data, i.e. cells that experience run and tumble motion and cells that experience extended undirected motion. Here again we build a possibility of switching between populations into the model with transition possibilities between slow diffusion, state 1, and tumble, state 2 as demonstrated in Figure 4.5 with the goal of this irreducible transition rate matrix converging to a near reducible transition rate matrix just as with the previous heterogeneous population models. This update gives us our new master equation

$$\frac{\partial P(t)}{\partial t} = \begin{bmatrix} -k_1 & k_2 & 0 & 0 \\ k_1 & -k_2 - k_5 & k_6 & 0 \\ 0 & k_5 & -k_3 - k_6 & k_4 \\ 0 & 0 & k_3 & -k_4 \end{bmatrix} P(t) \tag{4.7}$$

and the following SDEs for each state

$$dX = \begin{cases} \sqrt{2D_0}dW, & S(t) = 0 \\ \sqrt{2D_1}dW, & S(t) = 1 \\ \sqrt{2D_2}dW, & S(t) = 2 \\ v_3(\phi_t, \theta_t)dt + \sqrt{2D_3}dW, & S(t) = 3. \end{cases} \tag{4.8}$$

The results of the four-state model, given by equations 4.7 and 4.8, EM algorithm on our wild type data are consistent with out observations of a run and tumble sub-population and a diffusion sub-population. The results

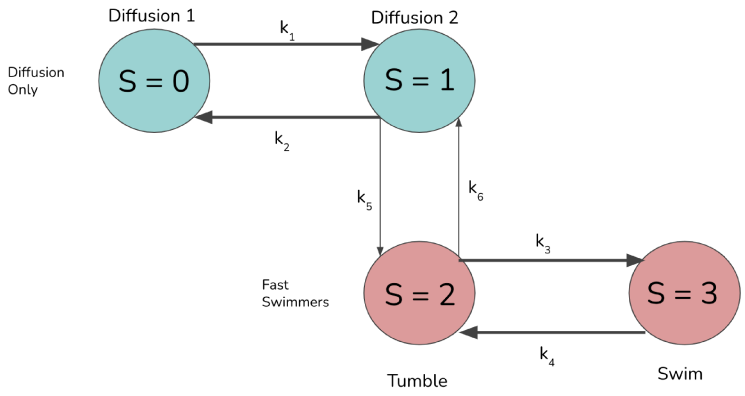

Figure 4.5: Four-State Model

of all parameter estimations can be seen in Table 4.5. With diffusivity values of $0.0339\mu m^2/sec$, $0.2705\mu m^2/sec$, and $0.4224\mu m^2/sec$ for each of the three undirected motion states, we see we have very clear and distinct states. Particularly with diffusion 1 and diffusion 2 which differ by an order magnitude.

Additionally, this heterogeneous model gives us the least number of tracks in both populations, with around 8% of tracks having at least one time point where the most probable state is from the diffusers population and one time point had the most probable state from the swim population. This is reflected in the fact that the transition rates between populations are significantly lower than the transition rates within the models. With parameter estimates consistent with what we expect, we decide to visualize the results with the data in order to further validate the four-state model, given by equations 4.7 and 4.8,. To do this, we turn to the probability distributions generated by the E-step of the EM algorithm.

We visualize the tracks as both states through time and overlaid the most probable state value onto the video data. By taking the maximum value from the E-Step distribution $P(S_t|X_{1:T}, \theta)$ for each time point, we are able to match

Parameter	Estimated Value
D_0	0.0339 $\mu m^2/sec$
D_1	0.2705 $\mu m^2/sec$
D_2	0.4224 $\mu m^2/sec$
D_3	1.1618 $\mu m^2/sec$
v_{mag}	31.723 $\mu m/sec$
k_1	0.6817 $transitions/sec$
k_2	1.4693 $transitions/sec$
k_3	0.1216 $transitions/sec$
k_4	2.7691 $transitions/sec$
k_5	.0361 $transitions/sec$
k_6	.0604 $transitions/sec$
$P(S_t = 0)$	0.59
$P(S_t = 1)$	0.27
$P(S_t = 2)$	0.05
$P(S_t = 3)$	0.08

Table 4.5: Parameter Estimations for the four-state Model (see Figure 4.5)

the most probable state of a cell at a specific time with the position data. We hope to qualitatively confirm the strength of our heterogeneous four-state model, given by equations 4.7 and 4.8, with these visualizations. By using this information and ImageTank, a program specifically designed to view and visualize scientific images, we are able to visually compare and contrast the homogeneous population model with the four-state model[35].

4.2.2 Visualization of Results

Just as we do with the three-state model, given by equations 3.13 and 3.14, in Figure 4.1, we visualize the tracks as states through time using the E-step predictions from our four-state heterogeneous population EM-algorithm. At first glance, a side-by-side comparison of the states through time from the three-state model, given by equations 3.13 and 3.14, and our four-state model,

given by equations 4.7 and 4.8, suggests the two models offer comparable state predictions. Both models predict that the undirected motion tracks experienced diffusion type motion, while the tracks that are directed motion tracks are predicted to experience swim motion. Additionally, we overlaid the state assignments onto the videos, in order to visualize the results of our EM algorithm dynamically through time. An example of this can be seen in Figure 4.7.

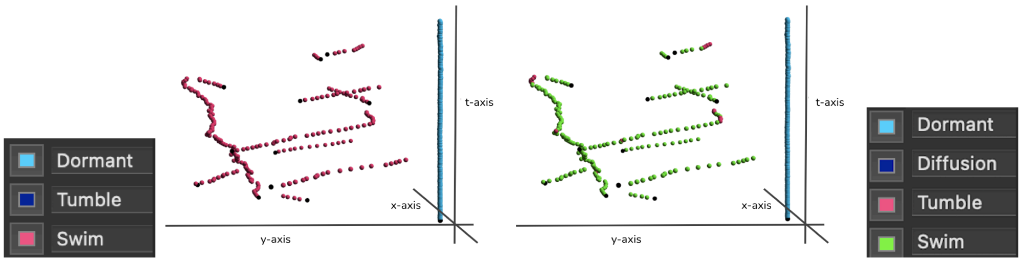

Figure 4.6: A side by side comparison in the state assignments from the three-state model (left) and four-state model (right) as well as their respective legends.

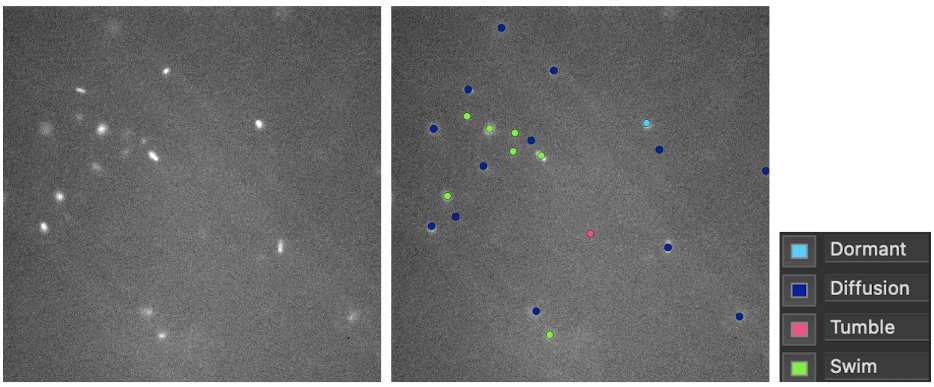

Figure 4.7: Left: a screenshot of a frame from a wild type microscopy video (see Figure 2.4). Middle: the same frame with state predictions overlaid. Right: Legend of state predictions.

One concern we had with the three-state model, given by equations 3.13 and 3.14, is that it appeared to overestimate the number of states in the

swim state. If we look at Figure 4.8, we can see a track experiencing directed motion with a change of direction but a tumble state is only present in the four-state model's, given by equations 4.7 and 4.8, predictions. We see time points we expect to be mapped into tumble mapped into swim consistently in the three-state model when it does not happen in the four-state model. We see these models disagree in a number of places, but given their disagreement in steady state approximations, that is to be expected. Notably, tracks which are mapped into the swim-tumble population in the four-state model, we often see the three-state model assign all time points to swim. This is interesting since the four-state model steady state estimations predict that 13% of cells are in the swimmers population (swim or tumble), which is the same number that the three-state steady state estimations predict in the swim state.

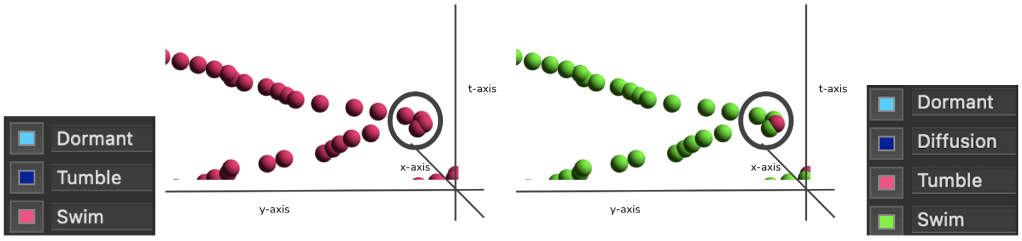

Figure 4.8: An instance of disagreement between the three-state model (left) and four-state model(right). The time instance in question is circled.

Through the visualizations we identify three main reasons a cell would transition between sub-populations: tracking errors, cells starting to experience directed motion, and uncertainty near the start and end of tracks. These three events can be seen in Figure 4.9, Figure 4.10, and Figure 4.11 respectively, and the implications of these events are explored in the next section.

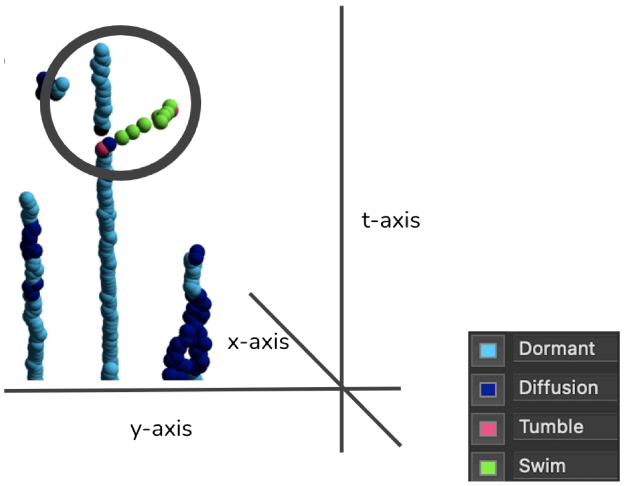

Figure 4.9: An example of a tracking error influencing state assignments. We see a track experiencing undirected motion being interrupted by a track experiencing directed motion. Instead of having the directed motion track as its own track, the tracking algorithm continues the original undirected track to include directed motion states while segmenting the remainder of undirected time points as their own track.

4.2.3 What does This Tell Us About the Motion of Wild Type Cells?

The two populations seen in our four-state heterogeneous population model, given by equations 4.7 and 4.8, are potentially phenotypically different sub-populations. One of the sub-populations represents cells that experience low amounts of directed motion while the other sub-population represents cells experiencing the classic run and tumble motion we expect. In addition to the expectations and parameter estimations, we are able to predict the steady state distribution based on the parameter estimations for transition rates. The results from the four-state model support what we know about the motion based heterogeneity experienced by *Salmonella*[31, 5]. However, there are a few explanations for the cause of the observed population heterogeneity.

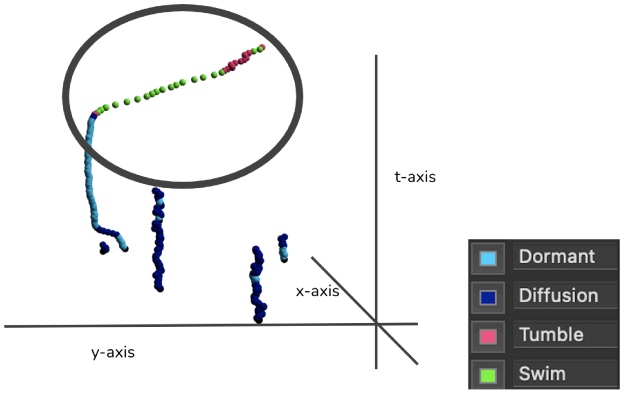

Figure 4.10: An example of a cell experiencing extended undirected motion and directed motion in the same track.

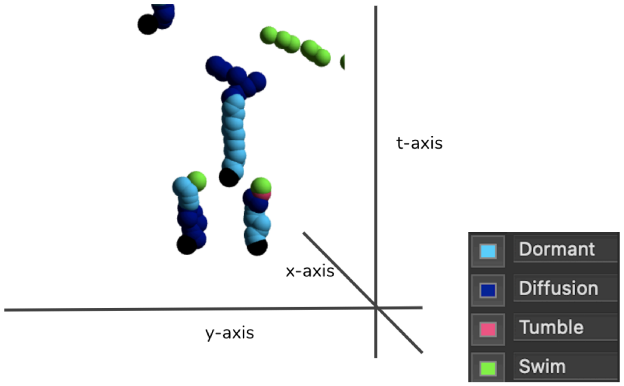

Figure 4.11: An example of uncertainty in state assignment for a short track.

Since run and tumble motion is well studied and understood, we know that the faster swimming population of cells has fully formed flagella which alternate between synchronized and un-synchronized rotation. The synchronized rotation of the flagella cause the *Salmonella* to have the forward directed run type motion. Similarly, the un-synchronized rotation of the flagella causes those same cells to tumble and reorient in space as each flagella creates its own directed force in opposition to the other flagella. A cell can not enter this population without the presence of motion structures, as the cell needs flagella

in order to experience the swimming motion. In this population, we observe diffusivity estimations from Table 4.5 of $0.4224\mu m^2/sec$ and $1.1618\mu m^2/sec$ in the swim and tumble states respectively, with a velocity magnitude of $31.723\mu m/sec$ in the swim state. In contrast, the three-state homogeneous model estimates the velocity magnitude for swimming to $19.743\mu m/sec$.

Additionally, we see from Table 4.5 that cells in the four-state model, given by equations 4.7 and 4.8, switch from run to tumble at a rate of 2.7691 $transitions/sec$ and tumble to run at a rate of 0.1216 $transitions/sec$. These transition rates are faster than the three-state model, given by equations 3.13 and 3.14,, who experienced rates of 1.5967 $transitions/sec$ for swim to tumble and 0.4125 $transitions/sec$ for tumble to swim, heightening our claim that the four-state model more accurately matches the video data, as we expect switches between tumble and swim to happen quite rapidly.

Since our models are "doubly stochastic", we also consider the steady state estimations as discussed in Section 3.1.2. To do this, we just need to solve the linear system $0 = W\rho$ where W is the transition rate matrix and ρ is a vector containing the steady state probabilities. Given a population of independent but identical individuals, as with our population, the steady state estimation can represent percentage of individuals in a state at a given time or percentage of time a track spends in a given state. In the case of the heterogeneous models, the former interpretation makes more sense, as it is not likely for a track to be in both slow and fast populations. Additionally, we can compare the steady state approximations to the observed percentage of increments spent in each state, to try to understand how the long term dynamics play out with our nearly irreducible transition rate matrix, with the only transitions happening

between the populations being $k_5 = 0.0361$ $transitions/sec$ and $k_6 = 0.0604$ $transitions/sec$ as seen in Table 4.5.

For the four-state model, given by equations 4.7 and 4.8, we see the steady state predicting 13% of cells in this sub-population, with 8% of all cells swimming. Because the matrix is nearly reducible, we did not expect the steady state calculations to fully match the observations, but we observe exactly 8% of time increments are spent in the swim state and 13% are spent in either the swim state or the tumble state per the E-step predictions. This matches the steady state predictions exactly. Interestingly, with the three-state model, given by equations 3.13 and 3.14,, our steady state predicts 13% of cells swimming with 13.5% of observed increments being in the swim state. This again points to the three-state homogeneous model overestimating the number of increments spent swimming compared to the four-state model. The state estimation generated by the three-state model also predicts 52% of cells are in the tumble state, which is a large discrepancy from the expected and observed steady states in the four-state model.

As we saw earlier, host immune cell pressures leads to the existence of cells in populations that lack motion structures[31, 5]. The lack of motion structures such as flagella would cause cells to experience diffusion-driven motion only. This diffusive motion can be observed in the sub-population with no swim state. In the four-state heterogeneous model, given by equations 4.7 and 4.8, this sub-population has two options for motion, the states we call diffusion 1 and diffusion 2 in Figure 4.5. Both of these states are classified only by diffusion-driven motion, with no directed motion. For these states, we see from Table 4.5 diffusivity estimations of $0.0339\mu m^2/sec$ and $0.2705\mu m^2/sec$

for diffusion 1 and diffusion 2 respectively. In comparison, dormant cells in the three-state model, given by equations 3.13 and 3.14, experience a diffusion rate of $0.0143\mu m^2/sec$, which again indicates that with the three-state model, more cells are being mapped to the faster motion states in comparison to the four-state model state prediction. The four-state model also predicts a rate of 0.6817 $transitions/second$ to switch from diffusion 1 to diffusion 2 and a rate of 1.4693 $transitions/second$ to switch from diffusion 2 to dormant 1.

Other possible reasons a cell would only experience the slower diffusion-driven motion states include cells which are in the process of developing flagella, cells which have flagella which are not rotating, or cells with a smaller number of un-synchronized flagella[17]. One question which is not easily answered is what would cause a cell to switch between the two diffusion states in this sup-population? One possible explanation is a cell synchronizes or unsynchronizes more flagella. That is, if a cell with 6 flagella has 3 rotating clockwise and 3 rotating counterclockwise could potentially experience a change in diffusion if it switched to 2 clockwise rotating flagella and 4 counterclockwise rotating flagella. Another explanation could potentially be that a cell is in the process of flagellar development, but does not fully develop their flagella enough to experience directed motion in our 30 second videos.

We can again turn to the steady state estimations for the four-state model, given by equations 4.7 and 4.8, for additional insights. These estimations place a total of 87% of cells in either the diffusion 1 or the diffusion 2 state. Additionally, we observe with the E-state expected values that 87% of time increments are in the diffusion 1 or diffusion 2 state, corroborating the steady state estimations. This suggests the vast majority of cells in our sample ex-

perience only undirected motion. In contrast, the three-state model, given by equations 3.13 and 3.14, only has steady state estimations of 34% of cells in the dormant state. This reflects that the three-state model predicts more cells in tumble and swim, reducing the diffusion and velocity estimators for all states, which is what we observe in the video data.

The fact that the four-state model, given by equations 4.7 and 4.8, predicts more cells experiencing undirected motion raises the question if lack of flagella is the only source of this distinct phenotypic heterogeneity? Can flagella turn off? Does a cell with one unsynchronized flagella experience a different diffusivity than a cell with multiple unsynchronized flagella? What type of motion is experienced when a cell is building flagella? How may cells from a given data set experience these various potential causes of the phenotypic heterogeneity? Unfortunately, the model cannot answer those questions since we observe only their motion, and we do not have biological evidence or technology to offer further insights. However, we can look at some of the more ambiguous results in order to attempt to gain additional insight into the observable phenotypic heterogeneity.

The cells which we see switch between the diffusion state and the tumble state stands as a third and unofficial sub-population. These cells interest us because they go against our assumption that two distinct sub-populations exist. These are the cells which have a high probability of at least one increment coming from the undirected motion population and one increment coming from the directed motion population. Various steps are taken to examine and identify these tracks. First, the array $P(S_t|X_{1:T}, \theta)$, given by equation 3.37, is investigated for each track. For each time point, the state number which gave

the maximum value in this array is recorded. Any track which has at least one time point where the maximum value is from the population experiencing undirected motion and one time point where the maximum value is from the directed motion population is recorded. This occurred in approximately 8% of tracks.

Similarly, the other probability generated by the E-step of the Expectation Maximization algorithm, the matrix $P(S_t, S_{t+1}|X_{1:T}, \theta)$, given by equation 3.38, is also examined. Again, for each track, the maximum value for every time point is recorded. Every track which had at least one time point with a most probable transition being from one sub-population to the other is recorded.

Next, the video location of every track that is flagged by this process of investigating posteriors is recorded. All such videos are then examined in order to investigate potential causes for switches between populations. To investigate this, we use ImageTank as described in Section 4.2.2 to visualize these tracks as states through time, but also overlaying the predicted state values onto the videos. These methods allow us to examine the ambiguous tracks directly. Additionally, we examine the actual probability values present in the two E-step results, as described in Section 3.2.4, in order to see if those offer any insights as well. This is a more indirect examination of the tracks. Through this process, we are able to answer some of our questions regarding these cells.

Given that our cells experience motion in three dimensions but the particle tracker works on a two dimensional video, occasionally tracking errors happen. If cells cross path in the third dimension, even though they do not

necessarily interact, the particle tracking algorithm can confuse their tracks. When this happens, occasionally the track appears to jump as it briefly goes behind another track in the third dimension or even change direction as the tracker confuses cells. These tracking error events are easily identified through visualizations and can be seen in Figure 4.9. Tracking errors are not common in the data at large but still worth noting as they make up the majority, at just under 5% of total tracks, of the third group.

Another reason some cells experience both directed and undirected motion arises when the cell starts moving with directed motion. This only appears to happen in less than 10 of the microscopy videos we have. Examples of the sudden switch from undirected to directed motion can be seen in Figure 4.10. The most logical reason for this is these cells are in the process of flagellar assembly or already has flagella which start off non rotational and become rotational. It has been documented that during the process of flagellar assembly, cells start off in a dormant like state and slowly start to rotate similarly to a tumble state, and eventually start swimming with directional motion[31]. However, if a cell has fully formed but not rotating flagella, starting to rotate those flagella would also cause the cell to start experiencing the directed motion pattern. Given that we only have the 2D microscopy videos and position data, it is impossible to tell if either of these phenomena are occurring in the case of our cells, but they both offer potential explanations for reasons our cells change from long periods of undirected motion to directed motion.

Finally, due to the stochastic nature of our model as well as the nature of using statistical methods for parameter estimations, switches in tracks between the undirected motion population and directed motion population sometimes

appear due to uncertainty. Specifically, near the start or end of a track, the confidence the EM algorithm has in probable states or probable transitions is reduced. These probabilities are given by the entries in the array given by equation 3.37 and the matrix given by equation 3.38. Because we use the most probable state for most of the analysis, it could mean that the most probable state has a 95% probability of being the the correct state or 40% probability of being the correct state. For those tracks which do not appear to have tracking errors or appear to be starting directed motion but are mapped into both populations, often a level of uncertainty arose in the tail ends. This occurred in about 3% of tracks, most of which are tracks that are short in length.

For the most part, we see the most probable state having a large probability of being the correct state while other states have near zero probabilities. However, toward the start and end of tracks, sometimes lower differences in probabilities arise, and when the algorithm becomes uncertain if a cell is in the tumbling or diffusion state, that is reflected in jumps between populations. An example of how these uncertainties manifest can be seen in Figure 4.11. Tracks which experience uncertainty in state assignments are unfortunately part of the nature of statistical parameter estimation tools and do not necessarily reflect the biology directly.

The four-state heterogeneous population model we propose for *Salmonella*, given by equations 4.7 and 4.8, achieves our goal of classifying sub-populations based solely on motion data. Not only are we able to properly identify multiple sub-populations in our data based on their motion, but due to the relevance of the model we are able to examine the biological implications of those sub-populations. By building our models from the run and tumble model type

which has been extensively studied, we are able to capture the natural movement of *Salmonella*. The four-state model identifies a sub-population which experiences traditional run and tumble motion as well as a sub-population that only experiences undirected, diffusion type motion. This division is potentially caused by a response to host immune cell pressure, as *Salmonella* fight to infect and survive. These bacterium experience a large amount of phenotypic heterogeneity, and using just the natural motion of these cells we are able to identify one facet of the heterogeneity.

Chapter 5

Conclusions and Future Work

By expanding an already existing three-state run and tumble model, we explored several larger models with the goal of capturing phenotypic heterogeneity observed in our dataset. We fit the model parameters to experimental data using the EM algorithm. The MLE transition rates naturally divided the four-state model into two submodels where switching between each submodel was very slow compared to switching between states within each submodel. The use of EM algorithm not only allowed us to fit the model parameters but also allowed us to make inferences about a given cell's motion state at a given time. The marginal posterior probabilities for the hidden motion state allowed us to investigate the nature of the observable motion and the extent of the heterogeneity. We found that in most cases, tracks displaying a non negligible probability of switching between the two sub-models could be explained by tracking errors and uncertainty stemming from short tracks.

The observable population heterogeneity was best captured by our four-state model, which included a sub-population of only undirected diffusive mo-

tion states and a sub-population of run and tumble motion. It is reasonable to conclude that those cells that displayed run and tumble motion were able to engage in synchronous flagellar mediated directed motion. However, the underlying physiological cause of switching between undirected diffusion states is less clear. Some possible explanations include complete absence of flagella, incomplete flagellar assembly, and inactivated non-rotational flagella.

In the future, we can expand on the results reported here to include uncertainty of the parameter estimations by using statistical methods such as sampling or variational Bayes. This will add additional credibility and context to the claims we make about the four-state model and motion heterogeneity.

In addition to calculating certainty of the parameter fits, we can see if the model can be applied to other flagella driven bacteria. In theory, just as the run and tumble type models can be applied to all bacteria of this type, the four-state heterogeneous population model could be applied to all bacteria of this type. That could shed light on whether other types of bacteria also repress flagellar growth in order to evade host immune cells (and other contributions to motion heterogeneity).

Similarly, we can also generalize the model to account for additional heterogeneity. In terms of flagella, cells could have no flagella, a cell could be building flagella, or a cell could have a number of fully formed flagella. Exploring how these different stages impact motion would be a large undertaking, due to the number of possible combinations. Additionally, *Salmonella* experience a wide range of phenotypic heterogeneity beyond presence or absence of motion structures, and we could investigate how things like metabolic rate potentially impact motion of our cells.

Bibliography

[1] Andreas J. Baumler, Renee M. Tsolis, and Fred Heffron. Contribution of fimbrial operons to attachment to and invasion of epithelial cell lines by salmonella typhimurium. *Infection and Immunity*, 64(5):1862–1864, 1996.

[2] Erika Gebel Berg. A new spin on the old gram stain. *Chemical and Engineering News*, 2015.

[3] Thibault Bertrand, Yongfeng Zhao, Olivier Benichoe, Julien Tailleur, and Raphael Voituriez. Optimized diffusion of run-and-tumble particles in crowded environments. *Physical Review Letters*, 120, 2018.

[4] David F. Blair and Howard C. Berge. Restoration of torque in defective flagellar motors. *Science*, 242(4886):1678–1681, 1988.

[5] Dirk Bumann and Oliver Cunrath. Heterogeneity of salmonella-host interactions in infected host tissues. *Current Opinion in Microbiology*, 39:57–63, 2017.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[7] Y. Dolak and T. Hillen. Cattaneo models for chemosensitive movement numerical solution and pattern formation. *Journal of Mathematical Biology*, 46:153–170, 2003.

[8] G. Fier, D. Hansmann, and R.C. Buceta. Langevin equations for the run-and-tumble of swimming bacteria. *Soft Matter*, 14:3945–3954, 2018.

[9] Food and Drug Administration. Fda.

[10] C.W. Gardiner. *Handbook of Stochastic Methods for Physics, Chemistry, and the Natural Sciences*. Springer-Verlag Berlin Heidelberg New York, 1985.

[11] Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):9–42, 2001.

[12] Daniel Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25), 1977.

[13] Xiangrong Xin Hans Othemer and Chuan Xue. Excitation and adaptation in bacteria–a model signal transduction system that controls taxis and spatial pattern formation. *Int. J. Mol. Sci*, 14, 2013.

[14] Tetsuo Iino. Genetics of structure and function of bacterial flagella. *Annual Review of Genetics*, 11:161–182, 1977.

[15] Malin E. V. Johansson, Jessica M. Holmén Larsson, and Gunnar C. Hansson. The two mucus layers of colon are organized by the muc2 mucin, whereas the outer layer is a legislator of host-microbial interactions. *Proceedings of the National Academy of Sciences*, 108(supplement_1):4659–4665, 2011.

[16] N.G. Van Kampen. *Stochastic Processes in Physics and Chemistry*. Elsevier Science Technology Books, 2009.

[17] Joyce Karlinsey, Shugo Tanaka, Vera Betternworth, Shigeru Yamaguchi, Winfried Boos, Shin-Ichi Aizawa, and Kelly Hughes. Completion of the hook-basal body complex of the salmonella typhimurium flagellum is coupled to flgm secretion and filc transcription. *Molecular Microbiology*, 37(5):1220–1231, 2000.

[18] Daniel Neamati Manuel Bedrossian, Marwan El-Kholy and Jay Nadeau. A machine learning algorithm for identifying and tracking bacteria in three dimensions using digital holographic microscopy. *Biophysics*, 5, 2018.

[19] Melvin Simons Michael Silverman. Bacterial flagella. *Annual Review of Microbiology*, 31:397–419, 1977.

[20] David A. Pegues; Samuel I. Miller. *Chapter 160: Salmonellosis*, volume 20e. 2001.

[21] Francesco Mori, Pierre Le Doussal, Satya N. Majumdar, and Gregory Scheh1. Universal properties of a run-and-tumble particle in arbitrary dimension. *Physical Review E*, 102, 2020.

[22] Shuichi Nakamura and Tohru Minamino. Flagella-driven motility of bacteria. *biomolecules*, 9(279):online, 2019.

[23] Jay Newby, Alison Schaefer, Pheobe Lee, M. Gregory Forest, and Samule Lai. Convolutional neural networks automate detection for tracking of submicron-scale particles in 2d and 3d. *PNAS*, 115(36), 2018.

[24] National Institude of Health. Wikipedia.

[25] E. M. Purcell. Life at low reynolds number. *American Journal of Physics*, 45(1):3–11, 1977.

[26] L. R. .Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, pages 4–16, 1986.

[27] Koen Martens Michiel Kleerebezem Peter Bron Sam van Beljouw, Simon van der Els and Johnnes Hoglbein. Evaluating single-particle tracking by photo-activation localization microscopy (sptpalm) in lactococcus lactis. *The Journal of Physical Biology*, 16, 2019.

[28] Holly Schroeder, Jay Newby, Alison Schaefer, Babu Subramani, Alan Tubbs, M. Gregory Forest, Ed Milao, and Samuel K. Lai. Lps-binding igg arrests actively motile salmonella typhimurium in gastrointestinal mucus. *Mucosal Immunology*, 2020.

[29] David W. Scott. *Statistics : a concise mathematical introduction for students, scientists, and engineers.* John Wiley Sons Ltd, 2020.

[30] Thomas J. Silhavy, Daniel Kahne, and Suzanne Walker. The bacterial cell envelope. *CSH Perspectives*, 2010.

[31] Ines Staes, Ioannis Passaris, Alexander Cambre, and Abram Aertsen. Population heterogenaity tactics and driving force in salmonella virulence and survival. *Food Research International*, 125, 2019.

[32] Sundharraman Subramanian and Daniel B. Kearns. Functional regulators of bacterial flagella. *Annu. Rev. Microbiol*, 73:225–246, 2019.

[33] Caressa Tsai and Brian Coombes. The role of the host in driving phenotypic heterogeneity in salmonella. *Trends in Microbiology*, 27(6):508–523, 2019.

[34] Andrea Villa-Torrealba, Cristobal Chavez-Raby, Pablo de Castro, and Rodrigo Soto. Run-and-tumble bacteria slowly approaching the diffusive regimes. *Physical Review E*, 101, 2020.

[35] Inc. Visual Data Tools. Imagetank.

[36] Ying-Ying Wang, Kenetta Nunn, Dimple Harit, Scott McKinley, and Samuel Lai. Minimizing biases associated with tracking analysis of submicron particles in heterogeneous biological fluids. *Journal of Controlled RElease*, 220, 2015.

[37] Chuan Xue. *Cell Movement, Modeling and Applications.*

[38] Chuan Xue. Macroscopic equations for bacterial chemotaxis: integration of detailed biochemistry of cell signaling. *Journal of Mathematical Biology*, 70, 2015.

[39] Chuan Xue and Xige Yang. Moment-flux models for bacterial chemotaxis in large signal gradients. *Journal of Mathematical Biology*, 73, 2016.

[40] Fuyan Wang Xin Wang Lin Wei Zhongjui Ye, Hua Liu and Lehui Xiao. Single-particle tracking discloses bindingmediated rocking diffusion of rod-shaped biological particles on lipid membranes. *Chemical Science*, 10, 2019.