# Multi-state inference applied to the analysis of noisy single-particle trajectories

by

Rebeca Cardim Falcão

B.Sc. Physics, Universidade Federal de Pernambuco, 2011

M.Sc. Physics, Universidade Federal de Pernambuco, 2014

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

**Doctor of Philosophy**

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Mathematics)

The University of British Columbia
(Vancouver)

December 2021

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

**Multi-state inference applied to the analysis of noisy single-particle trajectories**

submitted by **Rebeca Cardim Falcão** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in **Mathematics**.

**Examining Committee:**

Daniel Coombs, Department of Mathematics, UBC
*Supervisor*

Eric Cytrynbaum, Department of Mathematics, UBC
*Supervisory Committee Member*

Michael Gold, Department of Microbiology and Immunology, UBC
*Supervisory Committee Member*

Alexandre Bouchard-Côté, Department of Statistics, UBC
*University Examiner*

Daniel J. McDonald, Department of Statistics, UBC
*University Examiner*

Steve Pressé, Department of Physics and School of Molecular Sciences, Arizona State University
*External Examiner*

# Abstract

The single-particle tracking technique, where individual molecules are fluorescently labelled and recorded over time, is an important tool that allows us to study the spatiotemporal dynamics of subcellular biological systems at very fine temporal and spatial resolution. Mathematical models of particle motion are typically based on Brownian diffusion, reflecting the noisy environment that biomolecules inhabit. To detect changes in particle mobility within a trajectory, hidden Markov models (HMMs) featuring multiple diffusive states are commonly used.

In this thesis, we start by modifying a two-state hidden Markov model to take into account experimental errors and further improve the estimation of diffusion coefficients. In addition, we present a constrained hidden Markov model to analyze a specific set of experiments, where two fluorescence colours microscopy data is provided: molecules labelled at low density in one colour, and the second colour is molecules labelled at high density.

Hidden Markov models are typically specified with an *a priori* defined number of particle states, and it has not been clear how such assumptions have affected their outcomes. Here, we propose a method for simultaneously inferring the number of diffusive states alongside the dynamic parameters governing particle motion. We use the general framework of Bayesian nonparametric models and use an infinite HMM (iHMM) to fit the data. These concepts were previously applied in molecular biophysics. We directly extended iHMM models to the SPT framework and tested an additional con-

straint to accelerate convergence and reduce computational time. We tested our infinite hidden Markov model using simulated data and applied it to a previously analyzed large SPT dataset for B cell receptor motion on the plasma membrane of B cells of the immune system. We also incorporated experimental errors into this model, developing an algorithm that further improves the accuracy of parameter estimation, which we demonstrated using simulated data.

## Lay Summary

Understanding the spatial organization and underlying dynamics of proteins on the cell membrane is of great interest to biology since the cell membrane forms the physical boundary between the cell and its environment, mediating communication between the intracellular and extracellular environment. Focusing on experiments that track single proteins on the membrane, we developed methods to quantify the proteins' mobility by estimating their diffusivity. Diffusivity is a characteristic of random motion and tells us the average displacement of the motion over a time interval. It is common to use methods to classify protein mobility into different diffusive states capturing their heterogeneity. We further developed these methods to consider experimental errors in their classification. We also developed a framework that ascertains the number of diffusive states and estimates their diffusivity. Overall, these novel methods substantially advance our ability to analyze single-particle tracking data.

# Preface

This thesis consists of my original research, conducted in the Department of Mathematics at the University of British Columbia, Vancouver, Canada, under the supervision of Dr. Daniel Coombs. The following chapters contain previously published or in preparation work for which I was the principal investigator and author.

Chapter 3 is joint work with Daniel Coombs and has been published in the special issue of Physical Biology entitled *Advanced Methods for Reconstructing Molecular Dynamics from Single Cell Data* [34].

Parts of Chapters 2 and 5 are present work performed jointly with Daniel Coombs and Suzanne ten Hage. A manuscript is in preparation for submission to a scientific journal. I will share the first authorship with Suzanne ten Hage. I solely developed all algorithms that are used for the data analysis in the manuscript. I created the analysis pipeline in parallel with Suzanne. Some data presented in Chapter 5 is from David Depoil, who conceptualized and performed the experiments, and collected the data.

Experimental data analyzed in Chapters 2-4 are from previously published joint work with Libin Abraham and others in the laboratory of Michael Gold at the University of British Columbia [1].

# Contents

# List of Tables

# List of Figures

# Acknowledgements

First, I would like to thank my supervisor, Dr. Daniel Coombs, for his support, encouragement and patience. I have been very fortunate to have his valuable guidance during my Ph.D. studies. Thank you for all effort and time you put in to teach me about math, biology, and academic life in general. I am also grateful for your careful evaluations of all my writings, and your helpful feedback. Without your support, this thesis would not exist.

I would also like to thank my supervisory committee members: Dr. Michael Gold and Dr. Eric Cytrynbaum for all their support and helpful feedback throughout my Ph.D. years.

I am also thankful for my master's supervisor back in Recife, Dr. Fernando Parisio, and for Dr. Héctor Raúl Montagne Dugrós, whom I worked on my undergraduate research, for being great teachers and mentors.

Special thanks to my family and friends in Brazil and Canada, for all their support, and encouragement throughout the years. Equally important, for all the joy they bring to my life. Specifically: Ivan, Bela, Myrtw, Shaya, Thaís, Flávia, Mauro, Bruna, Eglânio, Florentino and Isa.

Last but not least, I thank Navid for all his love, patience, and support in everything. I also thank Yara, the cutest dog in the world, for all cuddles and pets that alleviated any stress during these last years. I also want to thank my parents, Ann and Derivaldo, for always putting us first in their lives. Lorena and Mirele, thank you for being these amazing sisters that

always inspired me. Finally, I want to thank Stella, Johnnie, Jack, Daniel, Bud, Samuel, Miller and Glenn for all the late nights writing this work.

# Dedication

Dedico essa tese para vovó Guinga que, da sua maneira, me mostrou que o riso tudo resiste, e a minha mãe Ann que, parafraseando Caetano, me ensinou à sempre pedir licença, mas nunca deixar de entrar.

# CHAPTER 1

# Introduction

Unravelling the complex network of molecules, their interactions and their connections with cell function has been a challenge of great interest in cell biology. A detailed understanding of the dynamics and organization of molecules on the cell surface is essential to understand the initiation of cell signalling and the processes of cell to cell communication [11, 123].

To study the dynamic processes of cells, we want to measure the mobility of molecules within the cells. Two common techniques to execute this measurement are fluorescence recovery after photobleaching (FRAP) and single-particle tracking (SPT). The principle of FRAP is by switching off fluorescent molecules, themselves labelling the molecules of interest, within a selected region of the sample [71]. Through high density labelling of molecules of interest with fluorophores, imaging of the sample over time is initiated. At this point, the image is a uniform bright field. Next, a region of the sample containing labelled molecules receives high intensity illumination. Upon receiving this high intensity light beam, the fluorescence of the labelled molecules rapidly decreases and they are removed from view. This phenomenon is known as photobleaching. Now the image on the micro-

scope consists of a uniform bright field with a dark spot. As times goes by, fluorescently labelled molecules diffuse throughout the sample reaching the dark spot, and replacing the bleached labels in the bleached region. After sufficient time, the image may become uniformly bright again. Using the intensity profile of the bleached region over time, one can calculate the diffusivity of the labelled molecules [71]. Due to the high density labelling, FRAP spatial resolution is limited (due to diffraction), and it provides an ensemble time average of the mobility of the molecules, rather than information on the motion of individual molecules. An alternative technique that combines high temporal and spatial resolution is single particle tracking. SPT has been extensively used to study important cell-surface molecules especially cell surface receptors [71, 106, 112, 117]. In this dissertation, I focus on developing new algorithms to analyse SPT data.

## 1.1 Single Particle Tracking

Single-Particle Tracking (SPT) is a powerful experimental tool that can provide information on each individual labelled molecules with nanometer-scale spatial precision on a milisecond timescale, depending on the labelling strategy, allowing the observation of their motion throughout the sample [110, 117].

As opposed to FRAP, which recovers the average motion of the particles during an experiment, SPT has a high spatial resolution revealing, on the scale of nanometers, the motion of individual particles [117]. This makes SPT an ideal tool to study heterogeneity of motion. SPT has been extensively used in studies of the lateral mobility of cellular membrane proteins [1, 23, 64, 144], intracellular protein motion [119, 143], colloidal physics [22], and microrheology [73, 138, 139]. The first SPT-like experiment was performed using gold nanoparticles with a diameter of 40 nm [26, 40, 71]. Initially called nanovid microscopy, this technique involved the attachment of gold nanoparticles to the biomolecules of interest, and the imaging relied on the phenomena known as Rayleigh scattering. Rayleigh scattering is the elastic scattering of light by objects much smaller than the wavelength of

the incident light. The Rayleigh scattering of the light by the gold nano particles has a high intensity, allowing for a very precise spatial localization of the biomolecules to the scale of nanometers [26, 40].

The stability of gold particles also allows for prolonged recording of the biomolecule dynamics. However, concerns were raised that their large size might influence the biomolecule's ability to interact with its surroundings and its motion as well. Therefore, there was a need for smaller probes. One could imagine decreasing the size of the gold nanoparticle. However, the intensity of the Rayleigh scattering decreases rapidly as the gold label is made smaller (as the sixth power of its size). In summary, gold nanoparticles are not ideal labels for tracking small biomolecules.

A solution to the size of the labelling objects is the use of fluorescence microscopy to detect individual molecules over time. The first implementation of fluorescence microscopy for tracking single particles was performed in 1993 [8]. After that, there was a surge of interest in fluorescence microscopy for SPT, and the development of new methods for data analysis that enabled new discoveries in biology [71].

### 1.1.1 Fluorescence Microscopy

Fluorescence microscopy refers to any imaging technique that uses fluorescent dyes to label the molecules of interest. Two commonly used fluorescence microscopic imaging techniques for single-particle tracking of cell membrane proteins are confocal microscopy and total internal reflection fluorescence microscopy [2, 37, 74, 76, 96].

Both of these techniques are based on imaging particles labelled with fluorescent dyes (fluorophores) [52]. Fluorescence is the emission of light in the visible spectrum by a substance that has previously absorbed radiation. Upon absorption of light, the molecules move into an excited state. To arrive in this state, the molecule is excited from the ground state $S_0$ to the excited state $S_1$ through the absorption of a photon of energy $h\nu_{exc}$, where $h$ is the Planck constant and $\nu_{exc}$ is the frequency of the absorbed photon. After a short time, it emits a photon of lower energy $h\nu_{emi}$, and relaxes to a

lower energy state. Since $\nu_{exc} \geq \nu_{emi}$, the wavelength of the emitting light is longer than the wavelength of the absorbed light. For example, if blue light is absorbed by the molecule, then a longer wavelength light such as green light would be emitted by the molecule. Figure 1.1 shows a sketch of the fluorescence process.

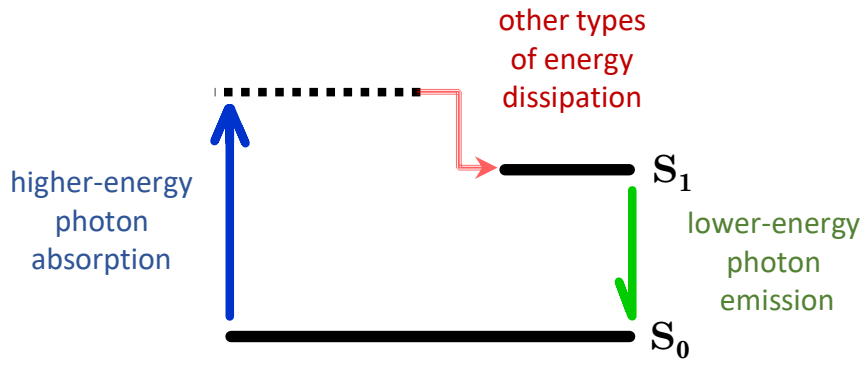



**Figure 1.1:** Energy transitions during fluorescence. $S_0$ indicates the ground state and $S_1$ indicates the excited state.

Most fluorescence microscopes are epifluorescence microscopes, where the incident and emitted photons pass through the same objective lens. This is achieved by the use of a dichroic mirror that reflects any light with a wavelength smaller than a threshold, and lets any light with a longer wavelength to pass through [71, 96, 105]. Thus, a laser emits light with wavelength $\lambda_{inc}$ that reflects on a dichroic mirror, and illuminates the sample, exciting it. Upon excitation, the fluorescent labels emit light with a wavelength $\lambda_{emit}$, which is larger than $\lambda_{inc}$. The emitted light passes through the dichroic mirror and reaches the detector[96]. A simplified scheme of epifluorescence microscopy is shown in figure 1.2.

As described above, an important advantage of fluorescence microscopy for the study of molecules on or in living cells is the ability to use small probes. However, there are some challenges and constraints of this technique. One of them is the large fluctuations in intensity (blinking), where the flu-

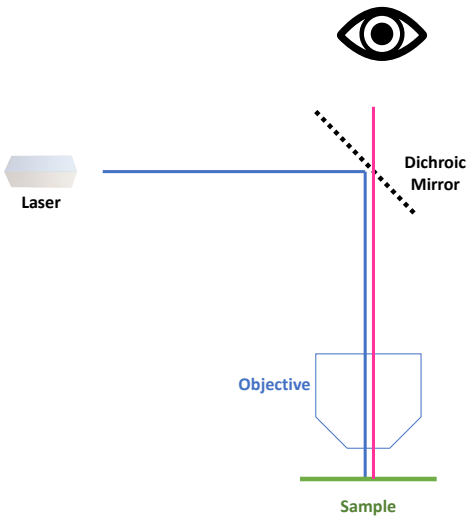



**Figure 1.2:** A simplified scheme of basic epifluorescence microscopy.

orescence label alternates between dark and bright states. Another one is the stability of the probes. Many types of fluorescent labels are not photostable and will cease to function after a period of activity. This phenomena is known as photobleaching, and the total number of photons that can be emitted varies between $10^5 - 10^6$ depending on the fluorescence molecule being used. Thus, the trajectory length of SPT labelled with fluorophores is on the order of seconds as opposed to minutes for gold particles [71]. FRAP utilizes photobleaching for measuring the diffusivity of the particles. This technique provides an ensemble measurement of diffusivity, given that there is no individual trajectory recorded, as explained above.

There are other alternatives labels that allow for the recording of longer trajectories similar to gold particles, such as quantum dots, beads, and recently developed fluorocube [88]. However, beads are as large as nanogold, resulting in similar concerns about their size influencing the molecule's ability to interact with their surroundings. Quantum dots are smaller than beads with diameters ranging from 2 nm to 10 nm. Therefore, they might influence the biomolecules' motion much less than nanogold particles or

beads. However, quantum dots blink, complicating the tracking process. See below for more detailed explanation. In the past year, a work published in Nature Methods introduced a novel probe called fluorocubes. They are small probes, $\sim 6nm$, that emit up to $\sim 43-$fold more photons than single organic fluorescent labels and have longer half-life [88].

In epifluorescence microscopy, the light propagates into the entire sample. The whole sample is illuminated by the focused light from the objective and also the out of focus light. This increases the amount of light reaching the cells and also may drive photobleaching. Given that light is toxic to cells, this increase contributes to the degradation of the cells. This is known as phototoxicity [105, 132]. Furthermore, fluorescence emission outside of the focal plane reaches the detector, reducing the signal to noise ratio in the image and its optical resolution. Other imaging techniques were created to tackle some of these limitations. We explain two of them below.

### Confocal Microscopy

An example of a fluorescence microscopy technique used in single-particle experiments is confocal microscopy. It is called confocal because a pinhole is placed at the optically conjugate plane of the focal plane. This pinhole allows only light produced by fluorophores very close to the focal plane to pass to the detector. This adaptation improves the optical resolution of the image, since out-of-focus light is largely eliminated [76, 96].

However, the signal-to-noise ratio is decreased, due to the decrease in intensity imposed by the use of the pinhole, since fewer photons arrive at the detector[76, 96]. Moreover, since the whole sample is excited by the light beam, photobleaching and phototoxicity limits the duration of image collection [76, 96].

### Total Internal Reflection Fluorescence Microscopy

A microscopy technique that limits these issues is called total internal reflection fluorescence microscopy (TIRF). The idea is to excite the sample using only an evanescent wave arising from the light beam, instead of the

beam itself. This decreases the intensity of the incident light on the sample, yielding a decrease in photobleaching and phototoxicity [2, 37, 74], and therefore, allowing for longer observations of living cells.

From Snell's law, we know that when a light beam passes from one medium to another it also changes its direction. Snell's law, also known as the law of refraction, is given by:

$$n_1 \sin(\theta_1) = n_2 \sin(\theta_2), \tag{1.1}$$

where $n_1$ and $n_2$ are the refraction indices of the media and $\theta$ is the angle of the incident light beam measured relative to the normal at the boundary between the two media. Thus, if the incident ray meets a medium with a lower refraction index, the refraction angle $\theta_2$ is larger than the incident angle $\theta_1$ (figure 1.3).

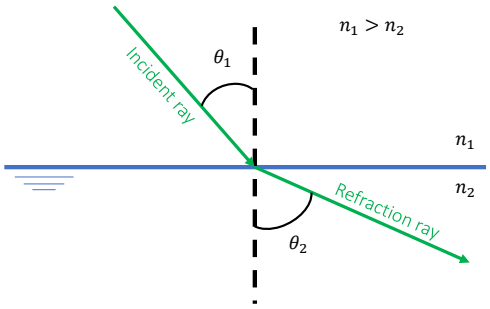



**Figure 1.3:** A sketch of the refraction law. The incident ray is refracted at the boundary with a medium of lower refractive index. From Snell's law (1.1), the refraction angle $\theta_2$ is larger than the incident angle $\theta_1$.

From equation (1.1), if $n_1 > n_2$, then we have $\sin(\theta_1) < \sin(\theta_2)$, and as we increase $\theta_1$, the refracted angle grows and eventually reaches a critical angle. This critical angle is the maximum angle where refraction still occurs. Reorganizing equation (1.1), we have:

$$\theta_{crit} = arcsin\left(\frac{n_2}{n_1}\sin\left(\theta_2\right)\right).$$

At the maximum incidence angle we have $\sin(\theta_2) = 1$. The critical angle is thus given by:

$$\theta_{crit} = arcsin\left(\frac{n_2}{n_1}\right).$$

For any incident ray with an angle larger than $\theta_{crit}$, the ray is totally reflected, and not refracted. Total Internal Reflection Microscopy uses this phenomenon to illuminate the sample. Figure 1.4 illustrates TIRF microscopy and epifluorescence microscopy and highlight their main differences. In epifluorescence microscopy, the incident ray is perpendicular to the sample, therefore illuminating and exciting all molecules on the sample. In contrast, TIRF incident light rays have angles larger than the critical angle, triggering the total internal reflection of the rays. Once the incident ray reaches the interface of the cover slip and the sample, it bounces back into the cover slip. However, a standing wave is created on the interface, allowing some of the incident light to penetrate a narrow portion of the sample near the coverslip. Through this wave, the continuity of the fields at the boundary is maintained. This wave is known as an evanescent wave and it is the source of the excitation of the fluorophores in the sample [2, 37, 74]. The intensity of the evanescent wave decreases exponentially with the distance from the interface, yielding a low penetration into the sample. Therefore, fluorophores closer to the interface are excited much more strongly than those further from the interface.

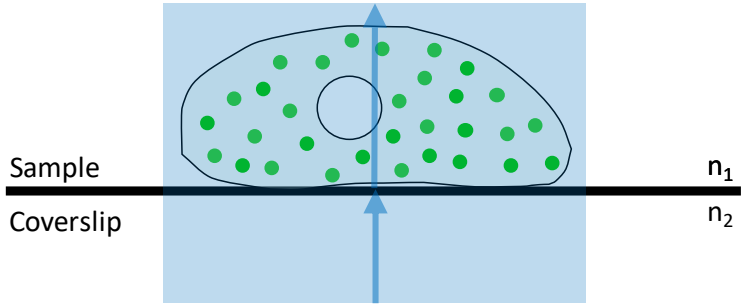
The penetration depth of the evanescence wave is given by:

$$d = \frac{\lambda}{4\pi}\left(n_1^2\sin(\theta_{crit}) - n_2^2\right)^{-1/2},$$

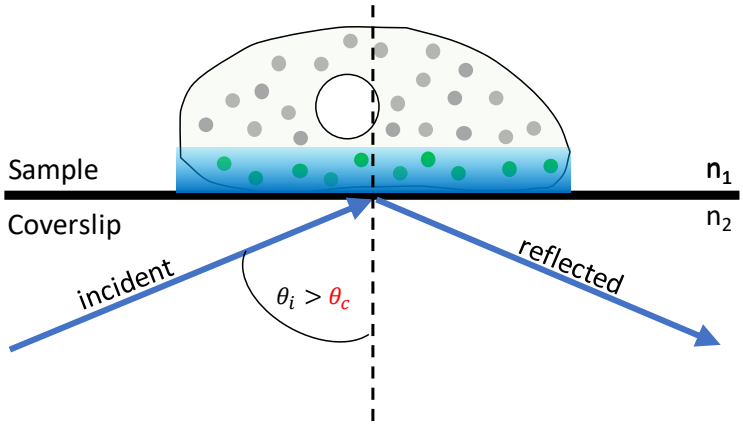
where $\lambda$ is the wavelength of the incident ray [74]. Typical values of $d$ using TIRF microscopy equipment is in the range of $100-200$ nm [37, 74, 74].

TIRF imaging exhibits decreased background noise, since fluorophores deeper within the sample are not excited. Moreover, the intensity of the light

reaching the fluorophores molecules is also decreased [37, 74], decreasing photobleaching (and phototoxicity).




(a)




(b)

**Figure 1.4:** Schematics of (a) Epifluorescence microscopy versus (b) TIRF microscopy

All the experimental data used in this work were acquired through TIRF

microscopy.

### 1.1.2 Detection and Tracking in Single Particle Tracking Experiments

The output of a single-particle experiment is a video where each frame is an image of the labelled molecules captured by optical microscopy. Figure 1.5 shows one frame of an SPT video. The output is analysed by image processing techniques to obtain trajectories of the labelled molecules. Different approaches to this problem have been developed over the last 20 years. The majority of approaches divide the problem into (i) detection and localization of labelled molecules in each frame of the video and (ii) linking the detected position sets from one frame to another [18, 22, 41, 51, 57, 110].

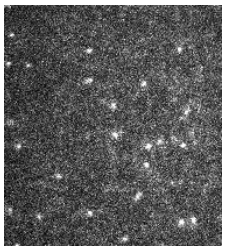


**Figure 1.5:** Screenshot of single-particle trajectories video.

Briefly, the identification of each molecule's location in each frame is the first step into obtaining all labelled molecule trajectories. Next, the positions of the molecules are linked to generate the final trajectories. A scheme of this framework is presented in Figure 1.6. First, we run a detection algorithm to detect particles in each frame. Next, a tracking algorithm links particles in different frames together, resulting in trajectories. Then, the processed data is obtained: a table with a frame index representing the time, and the x and y position of the particle at each time point.

During the linking step, we need to deal with molecules that transiently are not detected in one or more frames; and with molecules that show up or vanish permanently at some point. Both of these cause trajectories to split,

complicating the tracking procedure. Labelling probes that blink are more likely to generate these behaviours. These challenges are addressed by the tracking algorithm, which estimates the optimal trajectories from the set of localizations given by the detection algorithm. More details can be found below.

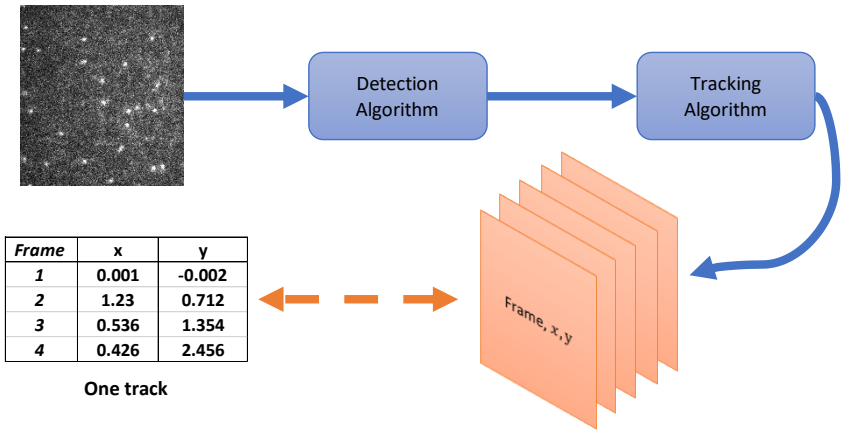



**Figure 1.6:** After acquiring the video of the labelled molecules, identification of the position of each molecule is performed in each frame of the video. Next, a tracking algorithm connects the positions of these molecules from frame to frame resulting in trajectories. The final output is a table, with information on the position for each frame and each object.

When light passes through a small aperture such as the pinhole of a camera, the shape of the light wave changes. This phenomena is known as diffraction. The different shapes of the aperture result in different patterns. For circular apertures, like the pinhole of a camera, the diffraction pattern is composed of concentric rings, alternating between bright and dark. The intensity of the rings decays as the distance from the center of the pattern decreases. Therefore, the central disk contains most of the intensity. This pattern was discovered by George Airy, and because of that is known as Airy pattern, and the central ring is called the Airy disk. The intensity of the Airy disk can be well-approximated by a 2-D Gaussian centred in the centre of the disk $(x_0, y_0)$ [71, 110, 132]. Then,

$$I(x,y) = I(x_0,y_0)\exp\left(-\frac{(x-x_0)^2}{2w^2}\right)\exp\left(-\frac{(y-y_0)^2}{2w^2}\right), \qquad (1.2)$$

where $w$ is the standard deviation of the spatial intensity distribution. This intensity profile is known as the point spread function (PSF). The detection algorithm typically consists of fitting the point spread function using equation (1.2). However, the Airy disk is an idealized description of the PSF that assumes a perfect optical system, where the light waves emerging from each point of the object and passing through the lens are convergent spherical waves. In actual optical instruments, the light waves surface have a more complicated form than spherical waves causing aberrations on the PSF [80, 89, 141], given that light rays from same points on the object do not meet on the same points in the image, leading to a blurring effect. The curved lens also causes PSF aberrations [80, 89, 141]. Furthermore, out-of-focus positions also affects the pattern of the PSF [80, 89, 141]. While it is out of the scope of this thesis to investigate these issues, they are a major problem for accurate localizations of single particles.

This standard deviation of the diffraction pattern for a point particle defines the resolution of the imaging system. The resolution is the minimal distance between two points where these two points can still be distinguishable, and depends on the wavelength of the light beam, and the numerical aperture of the system. For visible light, the maximum resolution is around 200 nm and is also called the diffraction limit. When fitting the Airy disk to the 2D Gaussian in equation (1.2), the localization precision of a single molecule can be improved to a few nanometers, a precision higher than the resolution of the microscopy system [71, 132].

The next step is to link the positions from frame to frame to build the trajectories of each labelled molecule. There are many possible trajectories when connecting dots from frame to frame. For example, let us say we have a 10-frame video, where in each frame there are 5 dots. Then, we need to choose among $5^{10} = 9765625$ trajectories. As we increase the number of particles, the number of possible trajectories gets larger, making it an intractable problem. However, if we further assume that all the 5 dots

are clearly spaced within a frame, then we could link the dots based on their distance between frames using a nearest neighbour kind of algorithm. Hence, the closest dot in the next frame is linked to a dot in the current frame. Figure 1.7 illustrates the connecting the dots process.

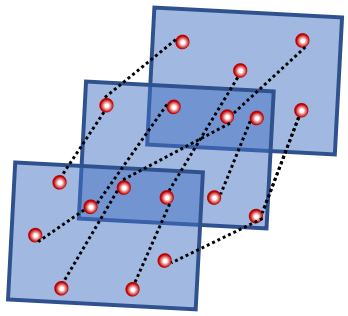


**Figure 1.7:** Linking the positions detected in each frame resulting on the labelled molecules trajectories.

If there are only a small number of labelled molecules in an experiment, this framework is usually feasible, given that the probability of two molecules crossing each other's paths is small. However, even at low density labelling there will still be molecules crossing paths. Moreover, particles can get closer than the diffraction limit (merging), and then move apart (splitting). Then, one of the two trajectories might be split, because one particle cannot be localized on a frame. Besides that, it is common for labelled molecules to transiently disappear and then reappear in different frames, making it harder to link the dots throughout the frames. To address these issues, additional conditions for linking the labelled molecules need to be added. There are many different algorithms to perform the linking between dots, and they are known as tracking algorithms [18, 71, 110, 111].

Each one of these algorithms utilizes different conditions for linking. Some of them join the detection and the tracking algorithms together to add more restrictions, and, as well as information on both the linking of the positions and on the detection of these positions. Others join the tracking step with the interpretation of the movements. In other words, they assume that the labelled molecules exhibit a specific type of motion or motions, and use this information to decide on the connection of the positions. These motions can be diffusion, directed motion, anomalous diffusion and so on. To this date there is no method that guarantees best performance accross different datasets, but there are best performing algorithm for each different dataset. The best performing algorithm depends on properties of the experiment, such as the label being used, and details of the experimental system in general [18].

In this work, we used a single software to do both detection and tracking process. The software used is called Icy and has a detection algorithm and a tracking algorithm built-in together [27]. Their detection algorithm is built from previous work [90], where the author developed a multiresolution algorithm based on undecimated wavelet transformation, and for their tracking algorithm [16, 17] they integrated the notion of target perceivability.

In Icy, a Multiple Hypothesis Tracking framework is used for tracking[16]. Instead of connecting the dots frame by frame, it uses information from future frames to decide the connection at frame $k$. From a set of possible associations between frame $k$ and frame $k + d$, the algorithm finds a subset that maximizes the likelihood of the associations ending at $k + d$. These associations are then clustered if they share at least one similar measurement. For each cluster, the associations with highest likelihood are kept. The final association is built by merging the associations for each cluster.

In this analysis, the likelihood is calculated using a Bayesian framework. First, we assume that the measurement of an object and its motion are not affected by other objects. The model also assumes the possibility of false detections. Moreover, it introduces the concept of perceivable state by the definition of a two state hidden Markov model (HMM). A track is perceivable if it can be measured. Therefore, a track is not perceivable if it has bleached,

physically disappeared or left the surveillance volume [16, 17]. Through this definition of these states, the algorithm allows for imperfections in the data, yielding a more robust algorithm. In practice, the Icy algorithm has been implemented to take advantage of parallel computing, allowing for faster performance[16]. The settings that I used for particle detection and tracking by the Icy software were: detection of bright spots over dark background with scale 2 and a threshold 70. There are three options for the scale, they specify the size of the spot one want to detect. Scale 2 is used for detection of spots of 4 to 7 pixels of diameter. The threshold is a parameter related to the wavelet method applied for the detection algorithm. If the threshold is set to 100, the set of detection found using the original algorithm is obtained. If this value is less than 100, then less detections are obtained, if this value is larger than 100, more detections are obtained. Diffusive and directed movements were considered in the tracking algorithm. We found that Icy is a fast and practical framework that delivers accurate trajectories.

Other tracking algorithms are also available such as u-Track [57] and Biggles [51]. Additional algorithms are described in reference [18], where the authors compared the performance of 14 different detection and tracking algorithms.

### Limitations of SPT

Although it is a very powerful tool, and perhaps, overall, the best method for probing the motion of individual molecules at millisecond to second time scales, there are some limitations to the single particle tracking technique.

As mentioned before, there are inherent limitations that arise from the probe being used to label the molecules, and the experimental settings.

1. Photobleaching of the probe leads to shorter trajectories, because the labelled molecules become invisible after a certain amount of time. This time depends on the probe and on the experimental set up. For organic probes, the higher the intensity of the incident light on the sample, the faster the photobleaching of the probe. This is due to the maximum number of photons that each fluorophore can absorb before

it goes dark. This number is usually around $10^5 - 10^6$ photons [71]. However, the lower the intensity of the incident light, the lower the signal-to-noise ratio, making it harder to detect the molecule. Quantum dots, beads, and nanogold are options for probes that generate long trajectories without compromising the signal-to-noise ratio.

2. Another limitation is the spatial resolution of the system. This depends on the wavelength of the incident light and the numerical aperture of the camera lens. For visible light, the maximum resolution is around 200 nm [71, 132]. To add context, isolated objects can be localized to $\sim$ 15-30 nm using fluorescent protein probes [1].

3. The finite spatial resolution is one of the reasons to apply low density labelling rather than labelling many molecules at once. The ability to distinguish labelled molecules in each frame is essential for good tracking algorithm performance. High density labelling increases the probability of labelled molecules being in close proximity (within the resolution limit), making it impossible to distinguish them. Moreover, it also increases the number of molecules crossing each other's path, resulting in merging and splitting, which makes it difficult to acquire the correct trajectories from the tracking algorithm [18, 71].

4. Another limitation is the time required to capture an image of the sample. This is quantified as the frame rate, the number of frames captured per second, and it depends on the speed of the detector as well as the time required to capture enough photons. Faster image acquisition implies fewer photons arriving at the camera and a smaller signal-to-noise ratio. Slower image acquisition impacts the study of fast dynamical processes. The temporal resolution of the data used in this thesis is 33 frames per second.

5. When molecules are moving too fast compared to the exposure time, the PSF of each molecule becomes indistinguishable, limiting the accuracy of the detection and tracking algorithm.

The decision on which probe to use has to be carefully taken, since they affect most of the aforementioned limitations. Gold nanoparticles or latex beads are very photostable, allowing longer trajectories (around 10 minutes) to be obtained, and yielding high spatial and temporal resolution. However, their size may greatly impact the motion of the biological objects being studied. Organic fluorescent molecules are much smaller probes and thus better suited for labelling molecules on a cell membrane. However, they photobleach quite rapidly and therefore generate trajectories that are typically seconds instead of minutes [71]. Quantum dots offer a compromise between size and trajectory length. Quantum dots are semiconductor nanocrystals whose wavelength emission is tuned by their size. Their size ranges from 5 to 8 nm, so they are at most twice the size of fluorescent molecule probes. They do not bleach and so allow for long trajectories to be captured. One optical drawback is the blinking. Quantum dots alternate between dark and bright levels of intensity, complicating the association steps and introducing tracking errors [1, 114]. Moreover, the size of quantum dots is still problematic. In previous work [1], we showed that cell surface proteins labelled using quantum dots had their mobility hindered. To demonstrate this, we performed single-particle tracking experiments on molecules labelled using quantum dots and organic fluorophores. Next, we quantified their mobility by estimating parameters using different models - diffusion process, multistate diffusion process, and confined diffusion process. We concluded that quantum dot probes still reduced the mobility of labelled cell surface proteins [1]. Therefore, the nature of the probe to be used must take into consideration multiple factors that may vary depending on the biological system. Figure 1.8 provides an illustration of a Cy3-labelled probe and a quantum dot probe taken from reference [1] with permission. The data from this work is used throughout this thesis.

## 1.2 Modelling of single particle trajectories

After the detection and tracking steps, we have the trajectories of the particles ready for analysis. We seek quantitative measures of the particle's

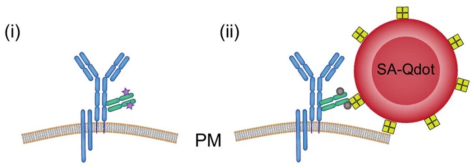



**Figure 1.8:** Sketch of (i) Cy3-labelled probe and (ii) quantum dot probe taken from reference [1] with permission under the Creative Commons License. To view a copy of the license visit: http://creativecommons.org/licenses/by/4.0/.

motion. There are many models one can fit to this kind of data, and the underlying system plays an important role in choosing the model.

### 1.2.1 Brownian Motion (Diffusion)

The simplest model describing the random motion of microscopic objects is Brownian motion (BM). BM is also known as diffusive motion and was first described in 1827 by Robert Brown who observed pollen immersed in water through a microscope. He observed that the particles of the pollen were following a random wiggling motion. In 1905, Einstein theorized that this random motion was due to collisions with water molecules that are themselves moving randomly due to thermal fluctuations. In the context of trajectories of proteins on the cell membrane, Brownian motion seems to be a well-suited model to describe the mobility of these labelled microscopic objects, i.e. nanoparticles subjected to thermal fluctuations from the molecules in the cell membrane and the surrounding semi-fluid environment.

Einstein derived an equation for the density of particles undergoing this random movement, $\rho(x,t)$. Instead of working with individual trajectories, he worked with the concentration of particles and how it changes over time and space [30].Let $\rho(x,t+\tau)$ be the density of these particles after some

increment $\tau$ at time $t$. We can relate the density at time $t+\tau$ with the density at time $t$ and on some spatial location $y$ such that $y = x + \Delta$, where $\Delta$ is the increment on space for each particle that happens during the time increment $\tau$. Assuming every movement of each particle is independent of the others, $\Delta$ has a different value for each particle. The density at time $t+\tau$ is equal to the integral of $\rho(x,t)$ weighted by the probability of having an increment $\Delta$ over all possible values of increment $\Delta$, therefore:

$$\rho(x, t+\tau) = \int_{-\infty}^{+\infty} \rho(x+\Delta, t)\phi(\Delta)d\Delta, \tag{1.3}$$

where $\phi(\Delta)$ is the probability density of an increment $\Delta$, thus $\int_{-\infty}^{+\infty} \phi(\Delta)d\Delta = 1$.

Expanding the left side of the equation over $\tau$ using Taylor series, and expanding the right side over $\Delta$, since both are very small we obtain:

$$\rho(x,t) + \tau\frac{\partial\rho(x,t)}{\partial t} = \int_{-\infty}^{+\infty}\left(\rho(x,t) + \Delta\frac{\partial\rho(x,t)}{\partial x} + \frac{\Delta^2}{2!}\frac{\partial^2\rho(x,t)}{\partial x^2} + \ldots\right)\phi(\Delta)d\Delta,$$

$$\rho(x,t) + \tau\frac{\partial\rho(x,t)}{\partial t} = \rho(x,t) + \frac{\partial^2\rho(x,t)}{\partial x^2}\int_{-\infty}^{\infty}\frac{\Delta^2}{2!}d\Delta, \tag{1.4}$$

$$\tau\frac{\partial\rho(x,t)}{\partial t} = \frac{\partial^2\rho(x,t)}{\partial x^2}\int_{-\infty}^{\infty}\frac{\Delta^2}{2!}\phi(\Delta)d\Delta, \tag{1.5}$$

where the integral of the first term of the Taylor series around $\Delta$ is equal to 1, and the second term vanishes due to integration on a symmetrical domain over a odd function. Defining the diffusivity constant $D$, as

$$\frac{1}{\tau}\int_{-\infty}^{\infty}\frac{\Delta^2}{2!}\phi(\Delta)d\Delta, \tag{1.6}$$

we obtain the diffusion equation:

$$\frac{\partial\rho(x,t)}{\partial t} = D\frac{\partial^2\rho(x,t)}{\partial x^2}. \tag{1.7}$$

The definition of $D$ shows the relationship between time and space increments. Moreover, we observe that $D$ is proportional to the mean of the

square of these increments. One can see how this is an important property of diffusion since we can measure the diffusivity of a system of particles just by using their displacements over time-defined increments.

To develop the theory of Brownian motion a little more, let's assume that at $t = 0$, all the particles are concentrated at $x = 0$, and they are moving on a one-dimensional system. Therefore $\rho(x, 0) = \delta(x)$. Let's also assume as $x \to \pm\infty$, $\rho(x, t) \to 0$. . Then, the solution of equation (1.7) (found via Fourier transform in x) that follows these boundary conditions is a Gaussian function:

$$\rho(x, t) = \frac{1}{\sqrt{4\pi D t}} e^{-\frac{x^2}{4Dt}}. \tag{1.8}$$

Notice that this model has only one solution. One can check that equation (1.8) is the solution by either plugging it into equation (1.7), or applying the Fourier transform to the initial condition, plugging it into equation (1.7), and solving the differential equation for the transformed function. After applying the inverse transform one recovers the Gaussian function as in (1.8).

Equivalently, the single particle probability distribution function obeys the same partial differential equation as in (1.7), and thus has the same fundamental solution (1.8).

Furthermore, the Brownian process is ergodic, meaning that the mean over time, i.e. $\langle X \rangle_t$, approaches the ensemble mean, i.e. $\langle X \rangle_X$, when $t \to \infty$. In other words, a sufficiently long sample of a Brownian process can represent the statistical properties of the process, or we can use many short samples of the process.

Using the ergodicity of Brownian motion and the definition of $D$, we can calculate $D$ from the trajectories obtained in the experiments. First, we calculate $\langle x^2 \rangle_x$ using (1.8) and obtain $\langle x^2 \rangle_x = 2D\tau$, where $\langle x^2 \rangle_x$ is the mean of the squared displacements (increments), and $\tau$ is the sampling time. Next, we can rewrite the average of the squared displacements over time instead of over samples. In our experimental trajectories, time is discrete, so we have a summation instead of an integral in the calculation of the mean. Therefore, we have:

$$\langle x^2 \rangle_t = \frac{1}{N-\tau} \sum_{t=1}^{N-\tau} \left( x(t+\tau) - x(t) \right)^2 . \tag{1.9}$$

This equation defines the mean square displacement (MSD) of the trajectory. Next, one can calculate the mean squared displacement curve for a set of trajectories. Varying $\tau$ gives different values inside the summation: $x(t-\tau) - x(t)$, and therefore a distinct value for $\langle x^2 \rangle_t(\tau)$. Therefore, the empirical MSD is a function of $\tau$ defining the MSD curve. For example, Figure 1.9 shows MSD curves for different types of processes. The curve that is linear with time is from a Brownian motion. Given that $\langle x^2 \rangle_x (\tau) = 2D\tau$ is also a function of $\tau$, and from the ergodicity of a Brownian motion, we know that $\langle x^2 \rangle_t = 2D\tau$. Therefore, the curve for the MSD trajectory should be linear with $\tau$, and the linear coefficient of the curve is equal to $2D$ for one-dimensional processes. More generally, the coefficient of the MSD curve is proportional to $2Dn$, where n is the dimension of the system. We can fit the empirical mean-squared displacement to a linear curve, and calculate the diffusivity from the slope. This was the basis of the approach adopted in several early works on SPT [64, 72, 99, 107, 109, 112].

Another approach to the mathematical theory of diffusion is obtained from the Langevin equation [10]. From Newton's second law, we know that the resulting force acting on an object is equal to $m\frac{dv}{dt}$, where $m$ is the mass of the object and $v$ is its velocity. For a particle in a fluid, the deterministic equation governing its motion is:

$$m\frac{dv}{dt} = -\frac{\partial V}{\partial x} - \eta v, \tag{1.10}$$

where the first term on the right hand side is due to external potential forces where $V$ is the energy potential due to external force $F$, and the second term is the drag term due to the viscous force on the particle, which is proportional to the velocity for tiny particles. Let's assume $V$ is linear with $x$, then $\frac{\partial V}{\partial x} = -F$.

For very small particles, for instance molecules on the cell membrane,

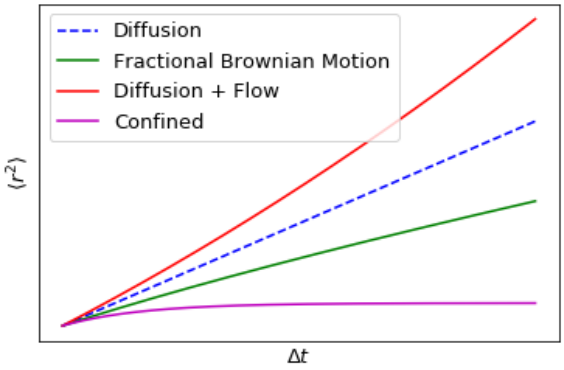



**Figure 1.9:** The mean-square displacement $\langle r^2 \rangle$ as a function of time $\Delta t$ for a diffusion process in a flow (red line), pure diffusion ( blue dashed line), a confined diffusion process (pink line), and a subdiffusive process called fractional Brownian motion whose MSD $\propto t^\alpha$ with $\alpha < 1$ (green line). In this graph, we set $\alpha = 3/4$.

the inertial effects are negligible. The Reynolds number is the ratio of the inertial forces to the viscous force acting on the object and is given by:

$$R = \frac{a\rho v}{\eta},$$

where $a$ is the dimension of the molecule, $\eta$ and $\rho$ are the viscosity and the density of the medium the molecule is moving around respectively. In the regime of low Reynolds number, the viscous force predominates over the inertial forces. This is the regime of molecules on the cell membrane, given their small size. In this regime, inertia plays no role. For example, if we are pushing a molecule to move it on the cell membrane, this molecule will stop rapidly once we stop pushing [98]. Give that inertia is negligible, we have $m\frac{dv}{dt} \simeq 0$ thus $0 = -\frac{\partial V}{\partial x} - \eta v$. Moreover, for particles that are not too big, the forces due to thermal fluctuations arising from the fluid molecules are not negligible, generating random forces that act on the particle. Thus, we have:

$$\frac{dx}{dt} = \frac{F}{\eta} + \xi, \tag{1.11}$$

where $\xi$ is a random variable representing the random forces generated by the thermal fluctuations of the fluid molecules. Equation (1.11) is known as the Langevin equation, and provides another approach to the diffusion equation. As time passes, the random forces applied on the particles average to zero over time, i.e. $\langle \xi \rangle_t = 0$. We also expect that the forces experienced over times longer than the time interval between collisions are uncorrelated, $\langle \xi(t)\xi(t') \rangle_t = A\delta(t - t')$, where $A$ is a constant. Integrating (1.11) with initial condition $x(0) = 0$, and supposing the potential is linear with time, we obtain: $X(t) = \frac{1}{\eta}Ft + \int_0^t \xi(t')dt'$, where $F$ is the external force (assumed to be constant). Averaging $X(t)$ and $\left(X(t) - \frac{1}{\eta}Ft\right)^2$ over the random forces, we have:

$$\begin{aligned}
\langle X(t) \rangle =& \left\langle \frac{1}{\eta}Ft \right\rangle + \left\langle \int_0^t \xi(t')dt' \right\rangle \\
=& \frac{1}{\eta}Ft, \qquad (1.12) \\
\left\langle \left(X(t) - \frac{1}{\eta}Ft\right)^2 \right\rangle =& \left\langle \left(\int_0^t \xi(t')dt'\right)^2 \right\rangle \\
=& \left\langle \int_0^t \xi(t')dt' \int_0^t \xi(t'')dt'' \right\rangle \\
=& \int_0^t \int_0^t \langle \xi(t')\xi(t'') \rangle dt'dt'' \\
=& \int_0^t \int_0^t A^2\delta(t' - t'')dt'dt'' \\
=& A^2 t. \qquad (1.13)
\end{aligned}$$

Since we know that the MSD is equal to $2Dt$, we find $A = \sqrt{2D}$. Moreover, $X(t)$ is a Gaussian process and its probability density follows a Gaussian distribution. For the case where there is no external potential, so $F = 0$, we recover equation (1.7), whose solution is the Gaussian function as in (1.8).

As diffusion is a stochastic process, we can also approach it through the probability density function $p(x,t)$ of position $x$ and time $t$. Using the fact that the diffusion process is Markovian (memoryless process), meaning that the next state of the process only depends on the current state, and the Langevin equation, we can perform a similar calculation to that presented above (1.7), to obtain the Fokker-Planck equation. The Fokker-Planck equation describes the evolution of the probability density through time and space:

$$\frac{\partial p(x,t)}{\partial t} = -\frac{\partial \mu(x) p(x,t)}{\partial x} + \frac{\partial^2 D(x) p(x,t)}{\partial x^2}, \tag{1.14}$$

where $\mu(x)$ is the drift. As written here, $D$ and $\mu$ might be functions of space as well. The previous equations are seen to represent simplified forms of the Fokker-Planck equation. Equation (1.14) is more general and describes the case where diffusivity and drift vary over time and space. Another way to obtain the Fokker-Planck equation is using the continuity equation $\partial p(x,t)/\partial t = -\partial J(x,t)/\partial x$, that represents the conservation of probability, where $J(x,t)$ is the probability current and is given by $J(x,t) = \mu(x)p(x,t) - \partial D(x)p(x,t)/\partial x$.

### 1.2.2 Confined motion

When we have a diffusive particle constrained to remain inside a domain, its MSD curve grows more slowly than the linear function, therefore indicating subdiffusive motion. As one can see in Figure 1.9, the MSD for a diffusion process constrained to remain within a domain is sublinear in time. Alternatively, the particle might be interacting with another particle or system and this interaction mimics a barrier that constrains the domain of the diffusive tracer. One example of such a system is the Ornstein-Uhlenbeck process. In this process, the motion of the particle is under the influence of random forces and an external force from a quadratic potential [10, 39]. This quadratic potential can be thought as a spring connecting the particle to the centre of the potential. Thus, the external force pulls the particle in direction to the centre of the potential, and the strength of the pull increases

with the distance of the particle to the centre. Suppose that the thermal fluctuations responsible for the random forces on the particle are large, then they can bring the particle to a position $x$ very far from the centre. However, as they move the particle on the space, the particle might get to a point where the restorative force is large enough, so that it is pulled back to the centre of the potential. Therefore, the particle is confined to this region. The Fokker-Planck equation for this process is given by:

$$\frac{\partial \rho(x,t)}{\partial t} = -A\frac{\partial \left(x\rho(x,t)\right)}{\partial x} + \frac{\partial^2 D(x)\rho(x,t)}{\partial x^2}. \tag{1.15}$$

We observe that the potential yields a linear drift term, that tends to restore the particle to the origin.

### 1.2.3 Anomalous Diffusion

There are other types of diffusive motion that display similar behaviour to the non-linear MSD curves shown in Figure 1.9. These other types are known generally as anomalous diffusion since one of the main properties of plain diffusion is that its MSD curve is linear with time. They are described as subdiffusive when their MSD curve grows sublinearly with time, or superdiffusive when the MSD grows faster than linearly with time. Anomalous diffusion can be thought as a generalization of diffusion when Einstein's assumptions are not valid. For example, if either the variance of the increment of each particle, i.e. $\Delta$, or the waiting time for the particle to move $\tau$ is not finite, a type of anomalous diffusion is generated [71, 78]. A commonly studied case supposes that MSD $\sim C(a)t^a$, where $a \neq 1$ and $C(a)$. There are many types of anomalous diffusion, each one with a different $a$ and $C(a)$ [77, 78]. As an example of anomalous diffusion, we have Levy flights, a superdiffusive process, whose step-length distribution follows a Levy distribution. The mean square displacement of a Levy flight is given by MSD $\sim t^{2/\alpha}$ for $\alpha < 2$, where $\alpha$ is a parameter of the Levy distribution. There has been a variety of evidence suggesting that many organisms (and cells) use a random search strategy that can be accurately modelled as a Levy flight [53, 55, 66]. A longstanding hypothesis is that Levy flight is the optimal random search

strategy. Intuitively, given that the variance of a Levy distribution is infinite, one can think that a particle undergoing a Levy flight would explore an area faster than a Brownian motion for example [56, 77, 78, 103, 104]. An example of subdiffusive motion is the fractional Brownian motion, whose mean square displacement is given by MSD $\sim t^{\alpha}$, where $0 \leq \alpha \leq 1$ a parameter of the motion. Fractional Brownian motion is used to model diffusive particles in a viscoelastic environment. In living organisms, viscoelasticity can be generated by crowding of molecules, or interactions among molecules [32, 63, 71, 77, 78].

### 1.2.4 Multi-diffusive states

A natural question is whether a microscopic particle obeying a random motion process might, in fact, transition between different processes as it explores its environment. For example, a cell surface receptor might encounter relatively large binding partners - leading to a transient reduction in its diffusivity. We therefore can seek to connect these models in a way that allows for transitions between different modes of mobility. One can do that using Markovian processes.

A discrete-time Markov chain (DTMC) is a stochastic process following the Markov property, i.e. the probability that a random variable $X$ is equal to $x$ at time $t$ given all the past values of $X$ is:

$$p\left(X_t = x \mid X_{t-1} = x_{t-1}, \ldots, X_1 = x_1\right) = p\left(X_t = x \mid X_{t-1} = x_{t-1}\right),$$

where $t$ is time and $X_t$ is the random variable value at time $t$. Therefore, a Markov chain is a memoryless process, where the value of the random variable at time $t$ depends only on the value of the random variable immediately before, at $t-1$ [39].

It is possible to construct a Markov model, such that every time point is associated with a state $\sigma_t$, and $\sigma_t = 1, 2, \ldots, K$, where $K$ is the total number of states. Moreover, each state out of $K$ possible states follows another process with a parameter $F_k$, for $k = 1, \ldots, K$, specifying that process. The probability of transition between states is defined through the

$K \times K$ transition matrix

$$T = \begin{bmatrix} p_{11} & \cdots & p_{1K} \\ \vdots & p_{kk} & \vdots \\ p_{K1} & \cdots & p_{KK} \end{bmatrix}.$$

For single-particle trajectories, the final data is the displacement of each particle for each frame as shown in Table 1.1.

| Time Interval | $\Delta x$ | $\Delta y$ |
|---|---|---|
| $t_1 - t_0$ | $x_1 - x_0$ | $y_1 - y_0$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $t_N - t_{N-1}$ | $x_N - x_{N-1}$ | $y_N - y_{N-1}$ |

**Table 1.1:** Experimental data of single-particle tracks experiments.

The Markov state chain underlying this data is hidden since one does not know in which state a particle is at each time point. As a simplifying assumption, we assume that the particle only changes states between successive frames. In other words, the particle exists in only one state per observed displacement. This is a reasonable approximation if the time step is sufficiently short compared to the transition rates. Therefore, we want to estimate the state parameters $F_k$ for each $k = 1, \ldots, K$, the transition matrix $T$, and the state sequence $s_i$ for $i = 1, \ldots, N+1$, where $s_i = 1, \ldots, K$ for the experimental dataset.

An example of "hidden Markov model" (HMM) applied to SPT analysis is given in the work by Das et al. [24]. In this work, the Markov chain allows for two states: one slow diffusive state and one fast diffusive state. The tracked proteins are membrane-associated proteins that could bind intracellular molecules, yielding a bound state with a slower diffusivity. When free, the molecule could attain faster diffusivity, being unbound and able to move on the membrane without restrictions. Das et al. developed an algorithm to estimate the diffusivity of both states, the transition probabilities, and the hidden sequence of states for each trajectory.

## 1.3 Parameter Estimation

The connection between the model and experimental data happens through the parameter estimation. This is where we acquire quantitative information from data. The estimation of parameters usually involves the optimization of a function. This function can be the likelihood of the model and its parameters given the data, or an approximation of this likelihood.

### 1.3.1 Maximum Likelihood Estimation

One method for estimating parameters is the maximization of the likelihood given an appropriate model. The most probable parameters that explain a particular dataset given the underlying model are the ones that maximize the likelihood.

The likelihood of a vector of displacements $\boldsymbol{\Delta x}$ is given by: $L\left(\boldsymbol{\Delta x} \mid \Theta\right) = p\left(\boldsymbol{\Delta x} \mid \Theta\right)$, where $p\left(\boldsymbol{\Delta x} \mid \Theta\right)$ is the probability of observing the vector of displacements $\boldsymbol{\Delta x}$ given the parameter set $\Theta$.

For a K-state Markov model, the conditional probability is given by:

$$p\left(\boldsymbol{\Delta x} \mid \Theta\right) = \sum_{\forall \boldsymbol{h}} p\left(\boldsymbol{\Delta x} \mid \boldsymbol{h}, \Theta\right) p\left(\boldsymbol{h} \mid \Theta\right), \tag{1.16}$$

where $\Theta$ is the set with the parameters describing each state's underlying process, $\boldsymbol{h} = (h_1, \ldots, h_N)$, for $h_i = 1, \ldots, K$, is the hidden sequence of states, $p\left(\boldsymbol{h} \mid \Theta\right)$ is the probability of having a sequence of states $\boldsymbol{h}$ given $\Theta$, and $p\left(\boldsymbol{\Delta x} \mid \boldsymbol{h}, \Theta\right)$ is the probability of observing $\boldsymbol{\Delta x}$ given the hidden sequence $\boldsymbol{h}$ and the parameters set $\Theta$. Since we are calculating the probability of observing $\boldsymbol{\Delta x}$ given the parameters of the model, we need to consider all possible sequences of states and sum over all of them. In other words, to compute the likelihood we need to sum over all possible sequence of states. For a $K-$state model and a data of size $N$, we have $K^N$ combinations of states for the hidden sequence of states. Therefore, direct calculation is computationally expensive.

An algorithm that calculates this likelihood efficiently is the forward

algorithm [4]. The forward probability is defined as the probability of observing the partial sequence of data $(\Delta x_1, \ldots, \Delta x_j)$ up to time point $j$, and being in state $i$ at time point $j$, with model parameters $\Theta$. This probability is given by:

$$\begin{aligned}\alpha_j(i) =& p\left(\Delta x_1, \ldots, \Delta x_j \mid h_j = i, \Theta\right) \\ =& \left[\sum_{k=1}^{K} \alpha_{j-1}(k) p_{ki}\right] p(\Delta x_j \mid h_j = i, \Theta).\end{aligned}$$

Then, the probability of observing $\boldsymbol{\Delta x}$ for a given parameter set is equal to:

$$p(\boldsymbol{\Delta x} \mid \Theta) = \sum_{i=1}^{K} \alpha_N(i).$$

The work in [24] uses this algorithm to maximize the likelihood and then estimate the parameters of the model. In this study, they have a two state model, therefore $K = 2$ in the above equations. Moreover, each state is described by a diffusion process. Then, the probability of the observed data given the parameters model is a Gaussian distribution: $p\left(\boldsymbol{\Delta x} \mid \boldsymbol{h}, \Theta\right) \sim \exp\left(-\frac{\Delta x^2}{2\sigma^2}\right)$, where $\sigma$ is the standard deviation of the diffusion process. From the Brownian motion section we know that the variance of a diffusion process is linear with time and equal to $\sigma^2 = 2dDt$ [10, 30, 39], where $d$ is the dimension of the process, and $t$ is the sampling time. For the experimental data in table 1.1, $n = 2$, where $t$ is the constant time interval for each displacement, so $t = t_1 - t_0, = t_2 - t_1 = \cdots = t_N - t_{N-1}$.

Besides estimating the parameters, it can also be important to estimate their uncertainty. One can achieve that using Markov Chain Monte Carlo (MCMC) algorithm to explore the likelihood and the vicinity of its peak, acquiring the posterior distribution of the parameters for each state. Statistics describing width of the posterior distribution around its peak can then be reported.

### 1.3.2 Model Selection

Maximum likelihood estimation methods provide a framework for estimating parameters given a model. Importantly, there is an underlying assumption that this model is the true model of the data. For the case of hidden Markov models, this model is defined by the number of states $K$, and the likelihood of the data is calculated based on a $K-$state Markov model, and its parameters set thus changes with $K$ as well.

Therefore, one needs to *a priori* ascertain the true number of states of a particular dataset, so that the parameters can be estimated. This decision could be based on some prior knowledge on the system being studied or through some heuristic properties of the data. However, this is usually not possible.

In living systems such as a cell membrane, a molecule can bind to other molecules in different ways, can move in a crowded environment, can move around in a free diffusive motion, can be attracted or repelled by different particles, and all of these situations affect its mobility [58, 67, 94]. Therefore, a molecule on a cell membrane possibly undergoes different modes of motion. This number of diffusive states not only indicates heterogeneity on the molecule motion, but also sheds light on the organization and structure of the biological system being studied. Besides being a biological unknown of great interest, the number of diffusive states is essential for the estimation procedure.

A clear need for a method that decides the best number of diffusive states based on data exists. Recently, work has been done to tackle this matter. One method was recently introduced by Linden and Elf [67], where a variational Bayesian approximation was used to estimate the parameters given the number of states of the Markov model and a variation of a cross-validation algorithm was implemented to select the best model among them, i.e. estimate the number of states. Another solution was described by Koo et al. [61], where ensembles of short particle trajectories were analyzed through an expectation-maximization approach to fit a Gaussian mixture model. The number of Gaussian mixtures is the equivalent to the number

of diffusive states. In this work, they calculated the likelihood of the data for a different number of Gaussian mixtures, and then used a Bayesian Information Criteria to decide the optimal number of Gaussian mixtures.

A common feature of both methods is the post-estimation procedures to select the best model: the estimation algorithm is applied for each possible model, and only afterwards is the best model selected via some criterion.

In two chapters of this dissertation, a different strategy to estimate the optimal number of states is developed. We aim to create a framework such that both the number of states and model parameters set are estimated simultaneously. This is done via a nonparametric Bayesian framework.

Bayesian nonparametric models are Bayesian models defined on an infinite-dimensional domain. They have the same structure as Bayesian models, and can be implemented by sampling the posterior distribution based on the prior and the likelihood. However, their parameter space is in principle infinite-dimensional. For the inference of Markov models, a Bayesian nonparametric approach allows for the number of states $K$ to be estimated alongside the other parameters, effectively, inflating the parameter space to infinity, since $K$ changes the size of the parameter set, and as $K \to \infty$ the parameter set cardinality becomes infinite as well [130].

## 1.4 Thesis Goals

There are many algorithms in the literature developed with the sole purpose of analyzing single-particle trajectories. Some of these algorithms consider only a single state diffusion process taking into account experimental errors [6, 79, 91, 99, 135], while others consider multiple state diffusive processes, different modes of motion [24, 122, 136], such as confined diffusion and free diffusion [28, 121], and so on.

In this thesis, we will present new algorithms to answer the following problems in SPT analysis:

- How does the lack of precision in experimental position measurements affect the parameter estimation for a multi-state diffusion process?

- How can one estimate the number of modes of motion, and their parameters simultaneously?

To answer the first question, we will present an algorithm that is a combination of the two-state hidden Markov model (as in [24]), and the covariance estimator that considers the experimental errors on the diffusion estimation developed by Berglund [6]. Next, we approach the second question by developing a nonparametric framework that estimates the number of diffusion processes and the model parameters simultaneously. Then, we apply this model to experimental data to check whether the number of modes is a fixed quantity throughout different runs of the same experiment. Finally, we implement this framework to take into consideration experimental errors similar to the two-state improved algorithm.

## 1.5 Thesis organization

This dissertation is organized by algorithms. In each chapter, an algorithm is described. They are ordered by complexity, and we can think of them as building blocks.

In Chapter 2, we describe the two-state hidden Markov model taking into account the measurement errors. This algorithm is an interpolation of two algorithms: the two-state hidden Markov model developed by Das et al. in [24], and a one-state diffusion estimator that takes into account the experimental errors developed by Berglund in [6].

In Chapter 3, the infinite hidden Markov model for single-particle trajectories is described. This model is a Bayesian nonparametric model. Therefore we also give a brief explanation of nonparametric Bayesian, Dirichlet processes and other tools that make this nonparametric framework possible.

In Chapter 4, we connect Chapter 2 and Chapter 3 and create a new algorithm, the infinite hidden Markov model for single-particle trajectories taking into account the experimental errors in position measurements.

In all three chapters, we validate the algorithm with simulated data and apply the algorithm to the same experimental data taken from [1].

Finally, in Chapter 5 we report on a hands-on experience of single-particle trajectories analysis. There, we developed a model specific to the experimental data being analyzed.

Figure 1.10 illustrates the relationships between the single-particle trajectories algorithms developed in this thesis and the concepts responsible for bringing the idea for their creation.

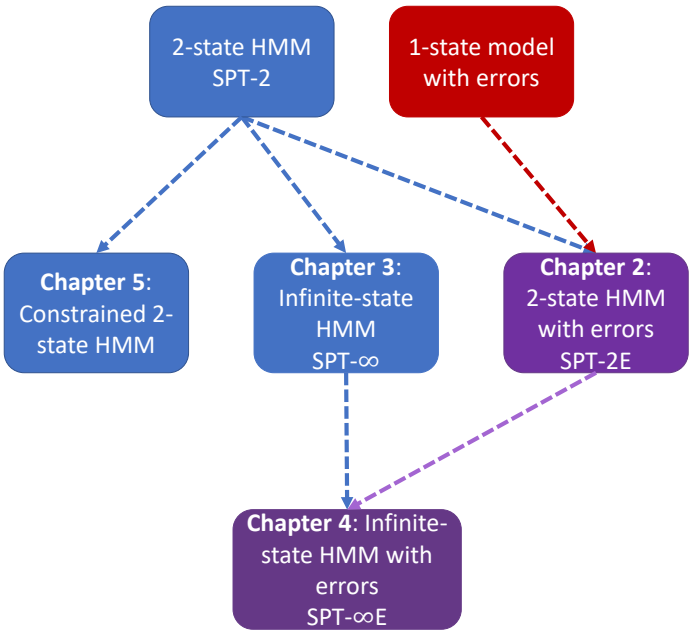



**Figure 1.10:** Relations among algorithms developed in this work and their predecessors. Algorithms were motivated by two previous works: the 2-state HMM [24] and the 1-state model with errors [6]. The two-state HMM is referred as SPT-2 throughout this thesis. In Chapter 2, we describe the integration of these algorithms into the 2-state HMM corrected for the experimental errors (SPT-2E). In Chapter 3, we generalize the 2-state HMM to a framework that allows for the estimation of the optimal number of states to best explain the data, together with their parameters. This is the infinite-state HMM, referred as SPT-$\infty$. Later, we include the experimental errors in this framework, and thus develop the infinite-state HMM with errors (SPT-$\infty$E). Lastly, in Chapter 5, we describe a pipeline created for a specific experimental setting, where we further develop the 2-state HMM into a Constrained 2-state HMM better suited to particular experimental settings.

# CHAPTER 2

## Two-State Hidden Markov Model considering Experimental Errors

Single-particle tracking has enabled us to achieve a better understanding of the mobility of molecules on the cell membrane, as well as their organization. SPT experiments generate individual trajectories of molecules with a high spatial resolution, and provide a much more detailed picture than other techniques like FRAP. These properties give us the possibility of studying the heterogeneity of the cell membrane (via heterogeneity of trajectories) as well as the ability to develop a more detailed mobility analysis that uses information from each particle's trajectory.

One reason for the mobility of microscopic particles on the cell membrane is due to thermal motion. Thermal motion is the random movement of a particle associated with its thermal energy. Brownian motion is the mathematical model used to describe the thermal motion of small particles in a fluid at a thermal equilibrium state in the absence of other effects (such as mobility constraints). Because of that, the most common model used to fit single-particle tracks of molecules on the cell membrane is the Brownian process, also known as diffusion.

The mean-square displacement, a measure of fluctuations of the particle's positions, of normal diffusion is a linear function of time, e.g. $\langle r^2 \rangle \sim t$, whereas in anomalous diffusion the MSD is a power function of the time, e.g. $\sim t^{\alpha}$. For $\alpha > 1$, we have superdiffusion, whereas for $\alpha < 1$, we have a subdiffusion. The majority of diffusion coefficient estimators rely on the mean-square displacement. An illustrative representation of that is on Figure 2.1..

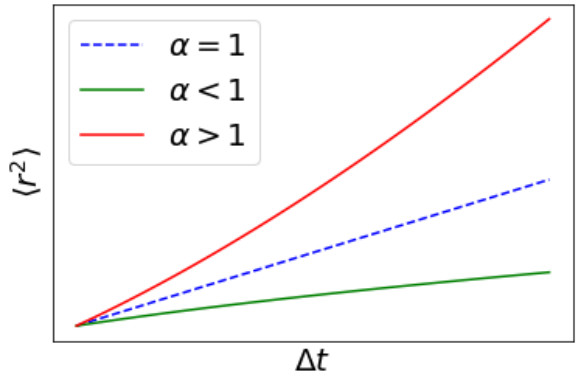



**Figure 2.1:** The mean-square displacement $\langle r^2 \rangle$ as a function of time $\Delta t$ for simultaneous diffusion and flow, pure diffusion, diffusion in the presence of obstacles, and confined motion.

However, a theoretical diffusion process might disagree with the data, because of experimental errors. From extensive previous work [6, 83, 99], we know that there are two common errors when generating single-particle trajectories: the static error, and the dynamic error. These two errors cause the displacements to be correlated, causing the MSD to cease to be linear in time. Therefore, when errors are not taken into account the estimated values of the diffusion coefficient may not be accurate. To address this issue, Berglund et al. developed a method to estimate diffusion coefficients of single-particle trajectories taking into account the static and dynamical error [6].

Deviations from linearity in the time versus MSD curve are very common,

and usually attributed to various phenomena, such as obstacles in the membrane, presence of flow, membrane compartmentalization, molecule binding and so on. Furthermore, since single-particle trajectories of proteins are commonly observed to be heterogeneous it is reasonable to assume the existence of transitions among different processes. This can be thought of as a molecule exploring the cell membrane, and, due to the heterogeneity of the membrane, this molecule may arrive in different environments, such as crowded ones, or may bind to other molecules, causing changes in its mobility. Because of that, a model that considers the possibility of a switch among processes (states) is more appropriate. As described in Chapter 1, Das et al. developed a two-state hidden Markov model to analyze single-particle trajectories [24]. However, experimental errors were not taken into account in this model. Here, we describe a new model composed of a two-state hidden Markov model that takes experimental errors into account. This model is based on the model of Das et al. [24], and Berglund's description of the measurement process [6].

In this chapter, we explain the measurement process for a particle with a switching diffusion coefficient, and the calculations needed to acquire the distribution of displacements. Next, we discuss how to estimate the parameters of the model. After that, we validate the algorithm using simulated data. The results are shown for different simulated data, and we find the range of parameters for which the algorithm works well. Finally, we show how to find the diffusive state of the labelled molecule at each time point of the data.

## 2.1 Measurement Process

As discussed above, the inclusion of experimental errors in the estimation of diffusion parameters should yield improved estimators. One of the main challenges is that measurement errors introduce correlations among the displacements of the particles, and that should be taken into account in the modelling.

First of all, we assume that the particle motion follows a Two-State Hid-

den Markov Model. As explained before, this model assumes that particles can switch between two states called 1 and 2, with probabilities $p_{12}$ to transit from 1 to 2, and $p_{21}$ to go from 2 to 1. Each of the two states represents different diffusion processes: say fast, and slow. For observed displacements $\Delta_1 \cdots \Delta_N$, we imagine a corresponding sequence $s_1, \cdots, s_N$, where $s_i = 1$ or 2 describing the state that a particle inhabits at each time point. If the positional measurements are subject to experimental errors this changes the diffusion process of each state.

In Figure2.2 we illustrate how the trajectories of a Brownian diffusion change when different errors are added to the process.

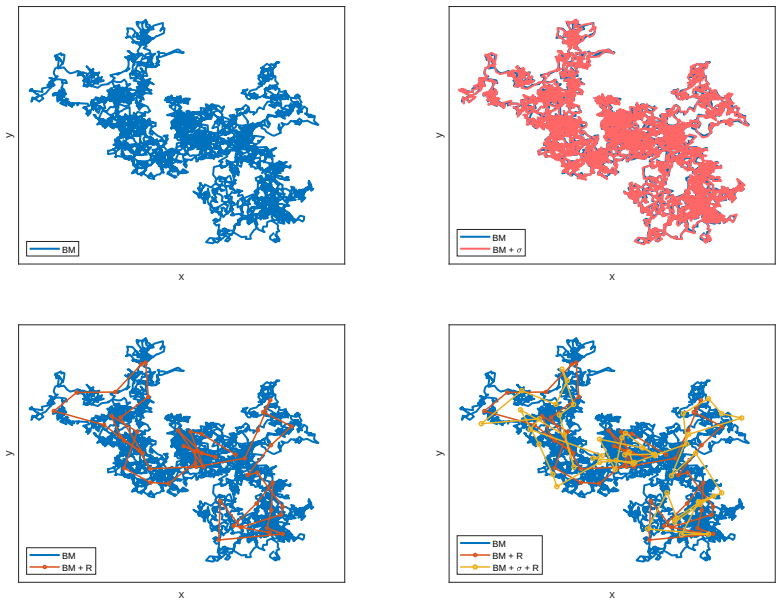



**Figure 2.2:** Trajectory of a simulation of one particle undergoing a two state Markov model with each state undergoing (a) a pure Brownian motion (blue curve), (b) a pure Brownian motion with static error (orange curve), (c) a pure Brownian motion with dynamical error (orange line with dots),(d) a pure Brownian motion with both static and dynamical error (yellow line). The pure Brownian displacements are the same in these simulations, the difference is the addition of experimental errors.

Let $Y(t)$ be the position of a particle undergoing a Brownian process. For simplicity here, imagine the particle moves in one dimension. When measuring the position of the particle, because of errors inherent in the experiments and observations, we do not acquire $Y(t)$ exactly. Instead, our measurements can be modelled as follow:

$$X_k = \int_{(k-1)\tau}^{k\tau} s\left[t' - (k-1)\,\tau\right] Y(t')dt' + \epsilon_k, \tag{2.1}$$

where $\tau$ is the time it takes to capture an image and $k$ is the image index. There are two types of noise considered here: the dynamical error, $s(t)$, and the static error, $\epsilon_k$ [6].

The dynamical error exists because the particle is moving while the image is captured, yielding a blur effect in the image. Therefore, the measured position is a weighted average of the true particle position, multiplied by the function $s(t)$. We call $s(t)$ the shutter function. This function represents the illuminating profile of the experiment, i.e. the relative amount of light received by the camera at each time point during the measurement time, $\tau$. The integral of $s(t)$ over the whole interval of the measurement is equal to one, i.e., $\int_0^\tau s(t')dt' = 1$. The static error is taken to be a Gaussian noise, given by $\epsilon_k$, with $\langle \epsilon_i \epsilon_j \rangle = \delta_{ij}\sigma^2$, representing the localization error in each frame.

Using the Brownian process property, and writing the starting time as $t_0$, we have:

$$\langle Y\left(t'\right) Y\left(t''\right)\rangle = Y^2\left(t_0\right) + 2D\left(min\left(t', t''\right)\right) min\left(t' - t_0, t'' - t_0\right), \tag{2.2}$$

we can calculate the average of the measured displacement distribution $\Delta_k = X_{k+1} - X_k$, and its covariance matrix. The covariance matrix is tridiagonal, reflecting that errors in one displacement also affect the previous and following displacements. Different than for a simple diffusion process, however, the matrix elements are affected by the underlying diffusivity at that point of the trajectory. We will now calculate the elements of the covariance

matrix.

### 2.1.1 Average

The average displacement is equal to:

$$\begin{aligned} \langle \Delta_k \rangle &= \langle X_{k+1} \rangle - \langle X_k \rangle, \\ &= X_0 - X_0, \\ &= 0. \end{aligned} \tag{2.3}$$

### 2.1.2 Covariance

The covariance matrix is given by: $\Sigma_{ij} = \langle \Delta_j \Delta_i \rangle - \langle \Delta_i \rangle \langle \Delta_j \rangle$, therefore: because $\langle \Delta_k \rangle = 0$, $\Sigma_{ij} = \langle \Delta_j \Delta_i \rangle$.

For $i = j$, we have:

$$\begin{aligned} \langle \Delta_i \Delta_i \rangle &= \langle (X_{i+1} - X_i)^2 \rangle, \\ \langle \Delta_i \Delta_i \rangle &= \langle X_{i+1}^2 \rangle + \langle X_i^2 \rangle - 2\langle X_{i+1} X_i . \rangle \end{aligned} \tag{2.4}$$

Expanding each term in Equation 2.4, we get:

1. **First term:** $\langle X_{i+1}^2 \rangle$.

$$\begin{aligned} \langle X_{i+1}^2 \rangle &= \left\langle \int_{i\tau}^{(i+1)\tau} s\left(t' - i\tau\right) Y\left(t'\right) dt' \int_{i\tau}^{(i+1)\tau} s\left(t'' - i\tau\right) Y\left(t''\right) dt'' \right. \\ &+ \left. 2\epsilon \int_{i\tau}^{(i+1)\tau} s\left(t' - i\tau\right) Y\left(t'\right) dt' + \epsilon_j \epsilon_j \right\rangle \end{aligned}$$

Since $\langle \epsilon_i \rangle = 0$ for all $i$ and $\langle \epsilon_i \epsilon_j \rangle = \delta_{ij} \sigma^2$, we obtain:

$$\begin{aligned} \langle X_{i+1}^2 \rangle &= \left\langle \int_{i\tau}^{(i+1)\tau} s(t' - i\tau) Y(t') dt' \int_{i\tau}^{(i+1)\tau} s(t'' - i\tau) Y(t'') dt'' \right\rangle \\ &+ \sigma^2 \end{aligned}$$

Let

$$I_{i+1} = \left\langle \int_{i\tau}^{(i+1)\tau} s\left(t' - i\tau\right) Y\left(t'\right) dt' \int_{i\tau}^{(i+1)\tau} s\left(t'' - i\tau\right) Y\left(t''\right) dt'' \right\rangle.$$

Next, apply the change of variables $w = t' - i\tau$ and $v = t'' - i\tau$, to find:

$$\begin{aligned} I_{i+1} &= \left\langle \int_0^\tau \int_0^\tau s(w)s(v)Y\left(w + i\tau\right) Y\left(v + i\tau\right) dwdv \right\rangle, \\ I_{i+1} &= \int_0^\tau \int_0^\tau s(w)s(v) \left\langle Y\left(w + i\tau\right) Y\left(v + i\tau\right) \right\rangle dwdv. \end{aligned}$$

Since the average here $\langle , \rangle$, is over realizations, we can permute the expected value to the inside of the integral, and due to the linearity of the expected value operator, we obtain the expression above. Next, we use the Brownian property, Equation 2.2, and the fact that we can partition a Brownian process into independent Brownian processes, to obtain:

$$\begin{aligned} I_{i+1} &= \int_0^\tau \int_0^\tau s(w)s(v)Y^2\left(\left(i-1\right)\tau\right) dwdv \\ &+ 2\int_0^\tau \int_0^\tau min\left(w + i\tau - \left(i-1\right)\tau, v + i\tau - \left(i-1\right)\tau\right) \\ &\qquad s(w)s(v)D\left(\left(i-1\right)\tau \to \left(i+1\right)\tau\right) dwdv \\ &= Y^2\left(\left(i-1\right)\tau\right) \\ &+ 2\int_0^\tau \int_0^\tau s(w)s(v)\left[D\left(i\tau \to \left(i+1\right)\tau\right) min\left(w, v\right)\right. \\ &\qquad \left. + D\left(\left(i-1\right)\tau \to i\tau\right)\tau\right] dwdv \\ &= Y^2\left(\left(i-1\right)\tau\right) + 2D\left(\left(i-1\right)\tau \to i\tau\right)\tau \\ &+ 2D\left(i\tau \to \left(i+1\right)\tau\right) \\ &\qquad \int_0^\tau \int_0^\tau s(w)s(v)min\left(w, v\right) dwdv. \end{aligned}$$

Here, we define:

$$D = \begin{cases} D(0 \to \tau),\ 0 \leq t \leq \tau, \\ D(\tau \to 2\tau),\ \tau \leq t \leq 2\tau, \\ \vdots \\ D((N-1)\tau \to N\tau),\ (N-1)\tau \leq t \leq N\tau, \end{cases},$$

where N is the number of displacements, and $D(\cdot \to \cdot)$ gives the diffusion coefficient for each displacement in the data.

Therefore,

$$\langle X_{i+1}^2 \rangle = I_{i+1} + \sigma^2. \tag{2.5}$$

2. **Second term:** $\langle X_i^2 \rangle$.

   Following the same procedure as before, we end up with:

$$\langle X_i^2 \rangle = I_i + \sigma^2, \tag{2.6}$$

   where

$$\begin{aligned} I_i &= \left\langle \int_0^\tau \int_0^\tau s(w)s(v)Y\left(w + (i-1)\tau\right) Y\left(v + (i-1)\tau\right) dwdv \right\rangle \\ &= Y^2\left((i-1)\tau\right)^2 \\ &+ 2\int_0^\tau \int_0^\tau min\left(w + (i-1)\tau - (i-1)\tau, v + (i-1)\tau - (i-1)\tau\right) \\ &\qquad s(w)s(v)D\left((i-1)\tau \to i\tau\right) dwdv \\ &= Y^2\left((i-1)\tau\right) \\ &+ 2D\left((i-1)\tau \to i\tau\right) \int_0^\tau \int_0^\tau s(w)s(v)min\left(w, v\right) dwdv. \end{aligned}$$

3. **Third Term:** $2\langle X_i X_{i+1} \rangle$.

$$\begin{aligned}\langle X_i X_{i+1}\rangle &= \left\langle \int_{(i-1)\tau}^{i\tau}\int_{i\tau}^{(i+1)\tau} s\left(t'-(i-1)\tau\right)s\left(t''-i\tau\right)Y\left(t'\right)Y\left(t''\right)dt''dt'\right\rangle \\ &+ \langle \epsilon_i\epsilon_{i+1}\rangle,\end{aligned}$$

since $\langle \epsilon_i\epsilon_{i+1}\rangle = 0$, we have:

$$\langle X_i X_{i+1}\rangle = \left\langle \int_{(i-1)\tau}^{i\tau}\int_{i\tau}^{(i+1)\tau} s\left(t'-(i-1)\tau\right)s\left(t''-i\tau\right)Y\left(t'\right)Y\left(t''\right)dt''dt'\right\rangle.$$

Applying the change of variables $w = t' - (i-1)\tau$ and $v = t'' - i\tau$:

$$\begin{aligned}\langle X_i X_{i+1}\rangle &= \int_0^\tau\int_0^\tau s(w)\,s(v)\,\langle Y(w+(i-1)\tau)\,Y(v+i\tau)\rangle\,dvdw \\ &= Y^2((i-1)\tau) \\ &+ 2\int_0^\tau\int_0^\tau s(w)\,s(v)\,D((i-1)\tau\to i\tau)\,min(w,v+\tau)\,dwdv \\ &= Y^2((i-1)\tau) + 2D((i-1)\tau\to i\tau)\int_0^\tau\int_0^\tau s(w)s(v)wdwdv \\ &= Y^2((i-1)\tau) + 2D((i-1)\tau\to i\tau)\int_0^\tau s(w)wdw. \qquad (2.7)\end{aligned}$$

Finally, using Equation 2.4: $\langle \Delta_i\Delta_i\rangle = \left\langle X_{i+1}^2\right\rangle + \left\langle X_i^2\right\rangle - 2\langle X_i X_{i+1}\rangle$ we obtain:

$$\begin{aligned}\langle \Delta_i\Delta_i\rangle &= 2D((i-1)\tau\to i\tau)\tau + 2\sigma^2 \\ &+ 2D(i\tau\to(i+1)\tau)\int_0^\tau\int_0^\tau s(w)s(v)min(w,v)\,dwdv \\ &+ 2D((i-1)\tau\to i\tau)\int_0^\tau\int_0^\tau s(w)s(v)min(w,v)\,dwdv \\ &- 4D((i-1)\tau\to i\tau)\int_0^\tau s(w)wdw.\end{aligned}$$

Simplifying this expression, we finally obtain:

$$\begin{aligned}\langle \Delta_i \Delta_i \rangle &= 2D\left((i-1)\tau \to i\tau\right) \\ &\quad \left[\int_0^\tau \int_0^\tau s(w)s(v)min\left(w,v\right)dwdv - 2\int_0^\tau s(w)wdw\right] \\ &+ 2D\left(i\tau \to (i+1)\tau\right)\int_0^\tau \int_0^\tau s(w)s(v)min\left(w,v\right)dwdv] \\ &+ 2D\left((i-1)\tau \to i\tau\right)\tau + 2\sigma^2. \end{aligned} \tag{2.8}$$

Now, we do a similar calculation for $j = i+1$, then:

$$\begin{aligned}\langle \Delta_i \Delta_{i+1} \rangle &= \langle \left(X_{i+1} - X_i\right)\left(X_{i+2} - X_{i+1}\right)\rangle \\ &= \left\langle X_{i+1}X_{i+2} - X_{i+1}^2 - X_i X_{i+2} + X_i X_{i+1}\right\rangle.\end{aligned}$$

Calculating term by term again, we have:

1. **First term:** $\langle X_{i+1}X_{i+2}\rangle$.

$$\begin{aligned}\langle X_{i+1}X_{i+2}\rangle &= \left\langle \int_{i\tau}^{(i+1)\tau}\int_{(i+1)\tau}^{(i+2)\tau} s\left(t'-i\tau\right)s\left(t''-(i+1)\tau\right)Y\left(t'\right)Y\left(t''\right)dt''dt'\right\rangle, \\ &= \int_0^\tau\int_0^\tau s(w)s(v)\left\langle Y\left(w+i\tau\right)Y\left(v+(i+1)\tau\right)\right\rangle dwdv \\ &= Y^2\left((i-1)\tau\right) + 2\int_0^\tau\int_0^\tau s(w)s(v) \\ &\qquad D\left((i-1)\tau \to (i+1)\tau\right)min\left(w+\tau, v+2\tau\right)dwdv \\ &= Y^2\left((i-1)\tau\right) + 2D\left((i-1)\tau \to i\tau\right)\tau \\ &+ 2D\left(i\tau \to (i+1)\tau\right)\int_0^\tau s(w)wdw\end{aligned}$$

2. **Second Term:** $\left\langle X_{i+1}^2\right\rangle = I_{i+1} + \sigma^2$ as in Equation 2.5.

3. **Third Term:** $\langle X_i X_{i+2}\rangle$.

$$\langle X_i X_{i+2}\rangle = Y\left((i-1)\right)^2 + 2D\left((i-1)\tau \to i\tau\right)\int_0^\tau s(w)wdw. \tag{2.9}$$

4. **Fourth Term:** $\langle X_i X_{i+1} \rangle$.

$$\langle X_i X_{i+1} \rangle \quad = Y\left((i-1)\right)^2 + 2D\left((i-1)\,\tau \to i\tau\right) \int_0^\tau s(w) w dw. \quad (2.10)$$

Putting all these terms together (First-Second-Third+Fourth), the third and fourth terms cancel out. We obtain:

$$\begin{aligned} \langle \Delta_i \Delta_{i+1} \rangle \quad = \quad & -\sigma^2 + \left(2D\left(i\tau \to (i+1)\,\tau\right)\right) \\ & \left[\int_0^\tau s(w) w dw - \int_0^\tau \int_0^\tau s(w) s(v) min(v,w) dv dw\right] \end{aligned} \quad (2.11)$$

Now, set $j = i+2$. Thus, we have: $\langle \Delta_i \Delta_{i+2} \rangle = \langle X_{i+1} X_{i+3} - X_{i+1} X_{i+2} - X_i X_{i+3} + X_i X_{i+2} \rangle$. Following the same procedure as before, we know that $X_i X_{i+2} - X_i X_{i+3} = 0$ and $X_{i+1} X_{i+3} - X_{i+1} X_{i+2} = 0$. Therefore $\langle \Delta_i \Delta_{i+2} \rangle = 0$. Moreover, we have:

$$\langle \Delta_i \Delta_{i+m} \rangle = \langle X_{i+1} X_{i+m+1} - X_{i+1} X_{i+m} - X_i X_{i+m+1} + X_i X_{i+m} \rangle .$$

Then, we conclude that: $\langle \Delta_i \Delta_{i+m} \rangle = 0$ for all integer $m > 1$.

Therefore, the covariance matrix for a sequence of displacements $X_0, X_1, \cdots, X_N$ is a symmetric tridiagonal matrix, where the non-zero elements are given by: $\langle \Delta_i \Delta_{i-1} \rangle$, $\langle \Delta_i \Delta_i \rangle$, $\langle \Delta_i \Delta_{i+1} \rangle$.

$$Cov = \begin{bmatrix} \langle \Delta_{11} \rangle & \langle \Delta_{12} \rangle & 0 & \cdots & 0 \\ \langle \Delta_{21} \rangle & \langle \Delta_{22} \rangle & \langle \Delta_{23} \rangle & \cdots & 0 \\ 0 & \langle \Delta_{32} \rangle & \langle \Delta_{33} \rangle & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \langle \Delta_{N(N-1)} \rangle & \langle \Delta_{NN} \rangle \end{bmatrix}$$

Finally, the full form of the displacement distribution for this measurement process is given by:

$$P(\Delta_1 \cdots \Delta_N) \propto \frac{1}{\|Cov\|} \exp\left( -\frac{1}{2} \left[\Delta_1 \cdots \Delta_N\right] Cov^{-1} \begin{bmatrix} \Delta_1 \\ \vdots \\ \Delta_N \end{bmatrix} \right) \quad (2.12)$$

### 2.1.3 Solving the Integral Terms

Now, we solve the integral terms in the covariance matrix. Here, we are considering experimental settings where one leaves the shutter open during the whole procedure. This yields a uniform illuminating profile, and optimizes collection of light in low signal-to-noise ratio experiments. The shutter function is thus $s(t) = \frac{1}{\tau}$.

1. $R_1 = \int_0^\tau \int_0^\tau s(v)s(w)min(v,w)dvdw$:

$$\begin{aligned} \int_0^\tau \int_0^\tau s(v)s(w)min(v,w)dvdw &= \int_0^\tau \int_0^v s(v)s(w)wdwdv + \\ &\quad \int_0^\tau \int_0^w s(v)s(w)vdvdw \\ &= 2\left(\int_0^\tau s(v) \int_0^v s(w)wdwdv\right) \end{aligned}$$

Integrating by parts, with $dq = s(v)dv$ and $p = \int_0^v s(w)wdw$, yields:

$$\begin{aligned} R_1 &= 2\left(\int_0^v s(v')dv' \int_0^v ws(w)dw\Big|_0^\tau - \int_0^\tau \left(\int_0^v s(v')dv'\right) vs(v)dv\right) \\ &= 2\left(\int_0^\tau ws(w)dw - \int_0^\tau S(v)vs(v)dv\right), \end{aligned}$$

where $S(x) = \int_0^x s(x')dx'$.

Applying integration by parts again in each term above:

$$\begin{aligned}\int_0^\tau ws(w)dw &= wS(w)\Big|_0^\tau - \int_0^\tau S(w)dw \\ \int_0^\tau ws(w)dw &= \tau - \int_0^\tau S(w)dw, \end{aligned} \tag{2.13}$$

and,

$$\begin{aligned}\int_0^\tau S(v)vs(v)dv &= \left(S(v)v\right)S(v)\Big|_0^\tau - \int_0^\tau \left[S(v)S(v) + S(v)s(v)v\right]dv \\ \int_0^\tau S(v)vs(v)dv &= \frac{\tau}{2} - \frac{\int_0^\tau S(v)^2dv}{2}. \end{aligned}$$

Therefore,

$$R_1 = \tau - 2\int_0^\tau S(w)dw + \int_0^\tau S(v)^2dv.$$

For $s(t) = \frac{1}{\tau}$, $S(x) = \frac{x}{\tau}$. Then,

$$R_1 = \frac{\tau}{3}.$$

2. $R_2 = \int_0^\tau ws(w)dw = \tau - \int_0^\tau S(w)dw$ as in Equation 2.13. Then,

$$R_2 = \frac{\tau}{2}.$$

We can now write out the full covariance matrix, as follows: Cov =

$$\begin{cases} \langle \Delta_i \Delta_j \rangle = 2D\left((i-1)\tau \to i\tau\right)\frac{\tau}{3} + 2D\left(i\tau \to (i+1)\tau\right)\frac{\tau}{3} + 2\sigma^2, \text{ for } j = i, \\ \langle \Delta_i \Delta_j \rangle = 2D\left(i\tau \to (i+1)\tau\right)\frac{\tau}{6} - \sigma^2, \text{ for } j = i-1 \text{ or } j = i+1, \\ 0, \text{ otherwise.} \end{cases}$$

The measurement process is fully defined by Cov.

Now, we need to design an algorithm to estimate the parameters of the model from experimental data. Although it may be possible to simultaneously estimate the parameters of motion ($\{D_1, D_2, p_{12}, p_{21}\}$) along with

the parameters governing measurement error $\left(\sigma^2 \text{ or } \epsilon,\ R\right)$, we will simplify here and assume that both dynamical and static errors are known. Then, the vector of parameters to be estimated is: $\Theta = \{D_1, D_2, p_{12}, p_{21}\}$. Next, following the same methodology used in Das et al. [24], we use Maximum Likelihood Estimation method to estimate the parameters, with the Markov Chain Monte Carlo algorithm to maximize the likelihood.

The Maximum Likelihood Estimation Method aims to estimate the most probable parameters that explain the observed data. Under the assumption of the underlying model, here a Markov process with 2 states, where each state yields data that follows a measurement process (diffusion process with errors in the measurement), the likelihood is defined by reference to the observed data. The parameters that maximize this likelihood are sought.

In this work, we applied a MCMC algorithm to explore the parameter space, and sample the vicinity of the likelihood peak. We used the Metropolis-Hastings (MH) algorithm, where for each parameter, the algorithm draws a new parameter value from a Gaussian distribution centered at the previous value of the parameter with standard deviation $\sigma$. Parameter values that increase the likelihood are accepted, and those that decrease the likelihood are accepted with a probability proportional to the change in the likelihood.

The framework for the exploration of the parameter space is divided into three phases. The first phase is a burn-in phase, where the algorithm explores the parameter space broadly. During the burn-in, we iterate 10000 times. If the acceptance ratio is larger than 45%, then the standard deviation of the proposal distribution step is doubled and we iterate 2500 more times. Otherwise the burn-in ends. We allow the standard deviation to double six times, and after that the burn-in is assumed to be completed whether the acceptance ratio of 45% has been reached or not.

The second phase is to decrease the 45% acceptance ratio to 33%. An acceptance ratio of 33% for a four-dimensional parameter space indicates that the search has reached a region of high probability, and that the posterior for each parameter is being sampled effectively, including the tail. First, the standard deviation of the proposal distribution is decreased to 20% of

its previous value. Then, the algorithm goes into a loop, with exit only when the acceptance ratio is accepted in three consecutive steps. The acceptance ratio has to be in between 32.6% and 33.4% to be accepted. In each iteration of this loop, the MH algorithm runs 1200 times. Every time an acceptance ratio is denied, the standard deviation for the proposal distribution is rescaled. If the acceptance ratio is smaller than the target value (33%), the standard deviation of the proposal distribution is decreased, if it is larger, the standard deviation is increased. The reason is that if the algorithm is accepting more steps than it should, the search is too local, and therefore, the size of the jump in the parameter space needs to increase to allow for a larger region to be explored. If the algorithm is not accepting enough steps, it means the search region is too large, and it should concentrate more in a local region.

Finally, the last phase is the exploration phase, where the algorithm explore the vicinity of the likelihood's peak by iterating 120000 times.

This algorithm is less efficient than gradient-based maximization schemes, and depending on the dataset can take days to finish. However, the advantage of using MCMC is that it naturally estimates the uncertainty in the estimated parameters of the model, allowing for a credible interval, and is less liable to become stuck in a local maximum [24].

The likelihood of the parameter set $\Theta$ for the displacements $\Delta_1, \cdots, \Delta_N$ is proportional to the probability of displacements $\Delta_1 \cdots \Delta_N$ given the parameter set. In summary,

$$\mathcal{L}\left(\Theta|\Delta_1 \cdots \Delta_N\right) \sim P\left(\Delta_1 \cdots \Delta_N|\Theta\right).$$

This probability is a summation of multivariate Gaussian distributions as in Equation(2.12), in which each distribution represents one possible realization of the model. The many different realizations are due to the many possible sequences of states yielding a different covariance matrix. Since we have 2 states, we have $2^N$ possible state sequences. This number is usually so large, that it is computationally infeasible to directly calculate $P\left(\Delta_1 \cdots \Delta_N|\Theta\right)$. Therefore, we use a modified version of a dynamical pro-

gramming algorithm known as the forward algorithm to calculate the likelihood [4, 24, 97]. Since the covariance matrix is tridiagonal, the forward algorithm has to be adapted from the simpler version described in Das et al. [24]. In the next section, we describe the adapted forward algorithm.

## 2.2 Adapted Forward Algorithm

As explained in Chapter One, the forward algorithm uses the forward probability, $\alpha_j(i)$, to calculate the likelihood step by step. For the pure diffusion process, the covariance matrix of the distribution of the displacement is diagonal, and one can calculate the likelihood from the likelihood of each displacement [24]. However, here we have a tridiagonal matrix, and instead of calculating the likelihood at each displacement, we must calculate the likelihood of two displacements.

The forward probability $\alpha_j(i)$ is the probability of observing the partial sequence of displacements, $\boldsymbol{O} = \Delta_1 \cdot \Delta_2 \cdots \Delta_j$, where displacement $\Delta_j$ is in state $i$, given the parameters $\Theta$. Then, the likelihood for a sequence of displacements
$\Delta_0, \Delta_1, \cdots, \Delta_N$ is given by:

$$P(\mathbf{O}|\Theta) = \sum_{k=1}^{2} \alpha_N(k), \tag{2.14}$$

where

$$\begin{aligned} \alpha_j(i) &= P\left[\Delta_1 \cdot \Delta_2 \cdots \Delta_j, s_j = i|\Theta\right] \\ &= \sum_{m=1}^{2}\sum_{k=1}^{2} \alpha_{j-2}(k) p_{ks_j} p_{s_j s_{j+1}} P\left[\Delta_j \Delta_{j+1}|s_j = i, s_{j+1} = m, \Theta\right]. \end{aligned} \tag{2.15}$$

The transition probabilities are $p_{ks_j}$ and $p_{s_j s_{j+1}}$, and they represent the past transition, and the next transition (or in which state the particle came from, and which state is going to). We need to sum over all the possible combinations of previous and next states. Because we have two states here, there are four possible transitions to consider.

Finally, $P\left[\Delta_j \Delta_{j+1} | s_{j-1} = k, s_j = i, s_{j+1} = m, \Theta\right]$ is the conditional probability of having the displacements $\Delta_j, \Delta_{j+1}$, given the previous state, the current state, the next state and the parameters. This probability comes from a two-dimensional multivariate Gaussian whose covariance matrix is equal to:

$$\mathrm{Cov}_{i,i+1}(s_{i-1}, s_i, s_{i+1}) = \begin{bmatrix} \Delta_{i,i} & \Delta_{i,i+1} \\ \Delta_{i+1,i} & \Delta_{i+1,i+1}, \end{bmatrix}$$

and

$$P\left[\Delta_j \Delta_{j+1} | s_j = i, s_{j+1} = m, \Theta\right] = \frac{1}{det\left(Cov_{j,j+1}\right)} \exp\left( -\frac{1}{2} \left[\Delta_j \Delta_{j+1}\right] Cov_{j,j+1}^{-1} \begin{bmatrix} \Delta_j \\ \Delta_{j+1} \end{bmatrix} \right). \tag{2.16}$$

Note that $\mathrm{Cov}_{i,i+1} = \mathrm{Cov}_{i,i+1}(s_{i-1}, s_i, s_{i+1})$, highlighting the dependence of the covariance on the previous, current and next states. Therefore, we need to sum the conditional probability over all possible sets up of states for the previous and next steps, as described in Equation 2.15. We sought to validate our new algorithm using simulated data.

Equation (2.15) only accesses the probability of pairs of displacements starting with odd displacements. This method is an approximation of the full likelihood of a $N \times N$ covariance matrix by $\frac{N}{2}$ disjoint blocks of size $2 \times 2$. Then, we are assuming that the parameters maximizing the full likelihood should be the equal to the ones that maximize the disjoint blocks likelihood. This should be a good approximation if the observed data is large enough. Other possible solutions are the switching Kalman filter and the junction tree algorithm [85, 87].

## 2.3 Results

We first simulated trajectories with experimental errors, and then estimated their parameter set using the noise-corrected two-state hidden Markov model. We also estimate the parameters using the SPT-2 [24], and then compare the performance of both algorithms. In this thesis, we refer to the

two-state HMM algorithm as SPT-2, and the two-state HMM corrected for experimental errors algorithm as SPT-2E.

We generated simulated data similar to that used in Das et al. [24], which itself was similar to SPT data obtained from experiments where the receptor LFA-I was labelled using fluorescent beads and imaged at 1000 Hz [12]. To start with, we generated eight such data sets. Each data set is composed of 4 particle tracks, each with 5000 frames, with a frame integration time of 1ms, and parameters $\Theta = \{1\ \mu m^2/s, 0.1\ \mu m^2/s, 0.3, 0.1\}$, with one of the following values for the standard deviation of the static error $\sigma = 0\ \mu m$, $0.0001\ \mu m$, $0.001\ \mu m$, or $0.01\ \mu m$, and with or without dynamical error. We assume that, the dynamical error corresponds to an uniform illuminating profile, so $R_2 = 1/2$, and $R_1 = 1/3$. Later in this section, we also show results from alternative simulated data sets.

As explained previously, the MCMC algorithm uses a burn-in phase to make sure that it is exploring a high probability region of the parameter space. For a 4-dimensional parameter space, an acceptance ratio around 33% is a good indication that this region has been found [24, 43, 70, 100]. All of these aspects are taken into account in the implementation of the algorithm. In Figure 2.3, we show the log-likelihood for one of the datasets during the burn-in phase, and the diffusion coefficient and transition probabilities estimates.

In the log-likelihood plot for the burn-in phase, one can observe the increase of likelihood in the first 1000 iterations. After this point, we observe small fluctuations of the likelihood, but no large further changes. This is an indication of convergence of the algorithm. When the algorithm reaches the region with high likelihood, it will stay there, and explore that region. The same behaviour is observed in the diffusion coefficients and the transition probabilities as expected.

After the burn-in phase, we want to thoroughly explore the high probability region. In this way, we can achieve a more accurate distribution of the parameters. The reported confidence interval for each parameter is calculated from this distribution. To achieve this, we performed 120,000 MCMC iterations after burn-in was achieved. Figure 2.4 shows the plots

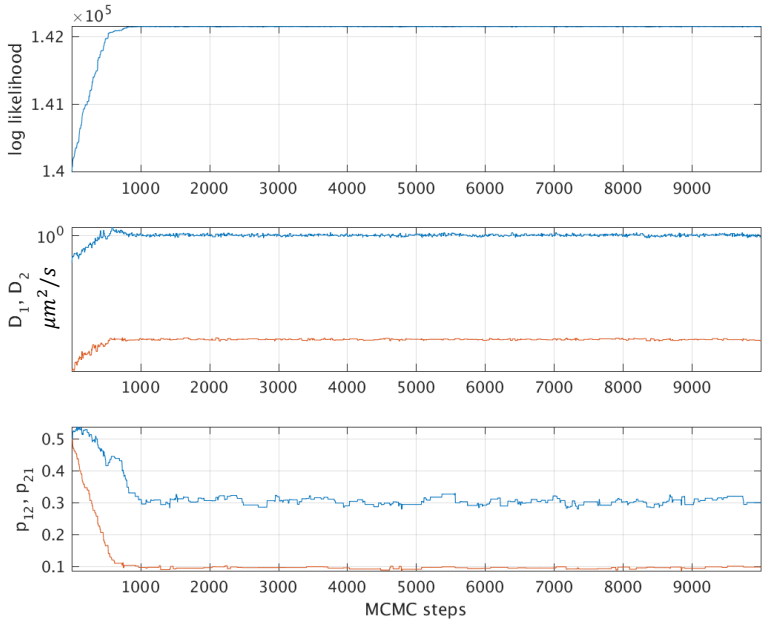



**Figure 2.3:** Algorithm convergence, showing the burn-in and steady phases. Graphs show values of the log likelihood for each iteration (MCMC steps), fast diffusivity values (blue curve) and slow diffusivity (orange curve) values for each iteration, and transition probabilities from fast to slow $p_{12}$ (blue curve), and from slow to fast $p_{21}$ (orange curve), for the first 10,000 MCMC steps.

of the log-likelihood, diffusion coefficients and transition probabilities value at each iteration. We observe small fluctuations around a similar value for each parameter, without any large change.

In Figure 2.5, we present the estimated distributions of the diffusion coefficients for each state. Results are shown for four different values of $\sigma$, the static error. Figure 2.5(a) shows results in the absence of experimental errors, and the others have the following values for the standard deviation of the error, $\sigma = 0\ \mu m$, $0.0001\ \mu m$, $0.001\ \mu m$, and $0.01\ \mu m$. We only considered static errors in these simulations. In Figure2.6, we have both types

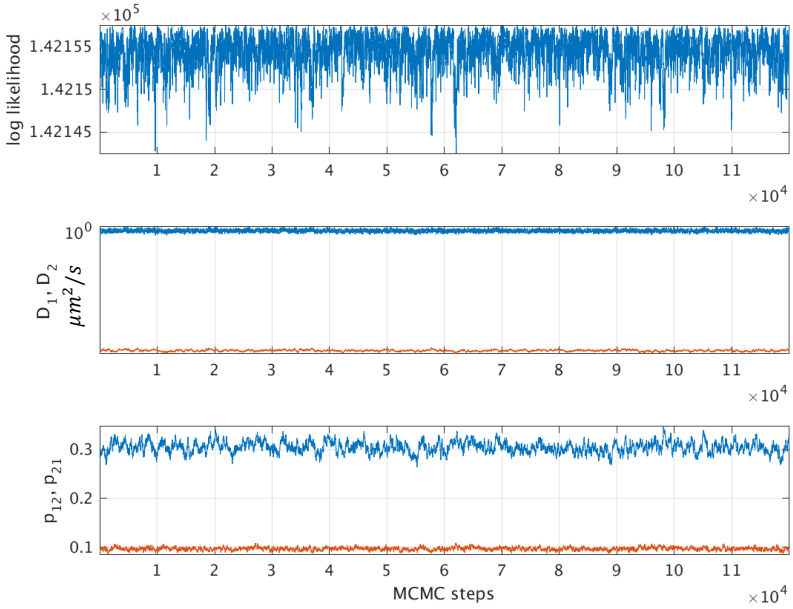
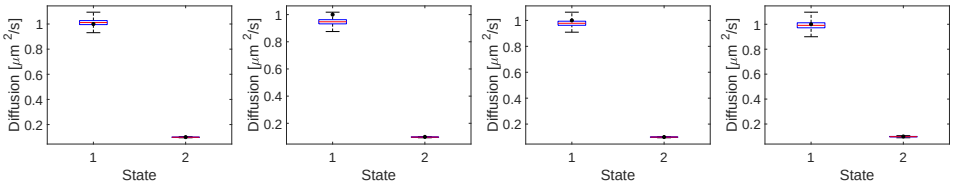



**Figure 2.4:** MCMC paths for the 120,000 iterations after the burn-in phase. Graphs show values of the log likelihood for each iteration (MCMC steps), fast diffusivity values (blue curve) and slow diffusivity (orange curve) values for each iteration, and transition probabilities from fast to slow $p_{12}$ (blue curve), and from slow to fast $p_{21}$ (orange curve).

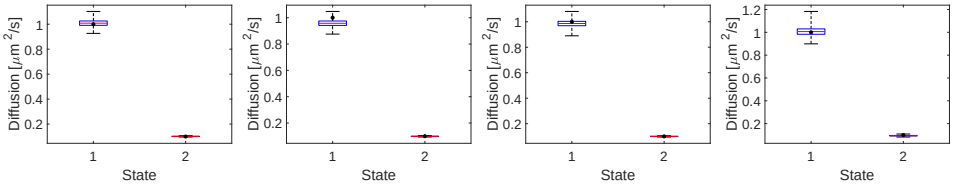
of experimental errors in the simulations, with the same values for the static error. From the graphs, we observe that the algorithm recovers the diffusion coefficients with roughly 1% to 5% error, even when $\sigma = 0.01\ \mu m$.

To compare the performance of the two algorithms, and whether it is an improvement to include the errors in the estimation, we also ran the two-state hidden Markov model without taking into account the errors on these noisy datasets. Table 2.1 shows the mean of the estimated diffusion coefficient of each state, and the mean of the estimated transitions probabilities for each dataset, using both algorithms. For the case with only static




(a) $\sigma = 0\ \mu m$ (b) $\sigma = 0.0001\ \mu m$ (c) $\sigma = 0.001\ \mu m$ (d) $\sigma = 0.01\ \mu m$

**Figure 2.5:** Estimated distributions of diffusion coefficient for each state of a simulated 2-state particle tracks with 5000 steps sampled at 1 $ms$ intervals, parameters $\Theta = \{1\ \mu m^2/s,\ 0.1\ \mu m^2/s,\ 0.5,\ 0.5\}$, and with only static errors whose standard deviations are the following:(a) $\sigma = 0\ \mu m$, (b) $\sigma = 0.0001\ \mu m$, (c) $\sigma = 0.001\ \mu m$, and (d) $\sigma = 0.01\ \mu m$.




(a) $\sigma = 0\ \mu m$ (b) $\sigma = 0.0001\ \mu m$ (c) $\sigma = 0.001\ \mu m$ (d) $\sigma = 0.01\ \mu m$

**Figure 2.6:** Estimated distributions of diffusion coefficient for each state of a simulated 2-state particle tracks with 5000 steps sampled at $1ms$ intervals, parameters $\Theta = \{1\ \mu m^2/s,\ 0.1\ \mu m^2/s,\ 0.5,\ 0.5\}$, and with both static and dynamical errors whose standard deviations are the following: (a) $\sigma = 0\mu m$, $R = 1/6$,(b) $\sigma = 0.0001\ \mu m$, $R = \frac{1}{6}$, (c) $\sigma = 0.001\ \mu m$, $R = \frac{1}{6}$, and d) $\sigma = 0.01\ \mu m$, $R = \frac{1}{6}$.

error, the SPT-2 estimates the correct parameters for all cases, except for the largest value of $\sigma = 0.01\ \mu m$, where the estimated value of the slowest diffusion coefficient is two times the true value, whereas the algorithm considering the errors recovers the parameters for all cases. For the simulated data with both experimental errors, the SPT-2 was not able to estimate the parameters correctly for any case. One can see that the estimated fast diffusion coefficients are equal to roughly half of their true value, while the slower diffusion coefficient estimates are smaller than their true value, except for when $\sigma = 0.01\ \mu m$. In this case, the slowest diffusion coefficient, $D_2$, is 1.5

times larger than its true value. Note that the static error generates anti-correlation in the data, whereas the dynamical error introduces a positive correlation in the displacements. Broadly speaking, they balance each other. Moreover, for the case with only static error, the failure of the two-state without errors happened only for the case where the standard deviation of the static error, $\sigma = 0.01\ \mu m$ was comparable to the standard deviation of a trajectory following the slow diffusive process, say $\sigma_2 = \sqrt{2D_2\tau} = 0.0141\ mum$. The estimate of the slowest diffusion was approximately two times the true value, suggesting that the variance of the estimated slowest Brownian process is approximately equal to the true variance plus the variance of the static error, $\sigma_2 \sim \sigma_{true2} + 0.01\ \mu m = 0.0141\ \mu m$. When both errors are included, the estimated value for the slowest diffusivity, which was doubled when the static error was included, was also affected by the dynamical error and was decreased by approximately one half, as one can see on Table 2.1, and the estimated value for $D_{slow}$ is $1.5D_{true2}$ instead of $2D_{true2}$ for the dataset with $\sigma = 0.01\ \mu m$.

Next, we simulated trajectories with different values for the diffusion coefficients and transition probabilities. We wanted to make sure the algorithm works for a wide different range of values, and to compare the performance of the algorithms. Table 2.2 shows the results of the estimation using both algorithms in these datasets. There, the simulated parameters were $\Theta = \{1\ \mu m^2/s,\ 0.2\ \mu m^2/s,\ 0.1,\ 0.05\}$, and all other parameters had the same values as in the previous simulated datasets. We can conclude that the SPT-2 applied on datasets including both static and dynamical experimental errors fails to recover any of the parameters. When only the static error is included, the uncorrected two-state model overestimates the diffusion coefficients, although this effect is only noticeable when the static error is large.

We also plotted the distribution of each parameter, to give an idea of the variance of the estimation, and to see how far the true value of each parameter is from these distributions. Figure 2.7 shows the plots for the estimated distribution of diffusion coefficients using the SPT-2E algorithm (a) on a dataset with standard deviation $\sigma = 0.001\,\mu m$ for the static error and no

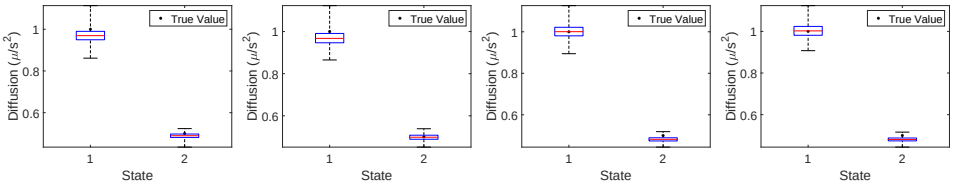
| *Parameters used in the simulation* | **No dynamical noise** | *1* | *0.1* | *0.3* | *0.1* |
|---|---|---|---|---|---|
| | $\sigma(\mu m)$ | $D_1$ $(\mu m^2/s)$ | $D_2$ $(\mu m^2/s)$ | $p_{12}$ | $p_{21}$ |
| SPT-2 | **0** | 1.0136 | 0.0992 | 0.2978 | 0.1017 |
| | **0.0001** | 0.9468 | 0.0983 | 0.2929 | 0.0974 |
| | **0.001** | 0.9798 | 0.1003 | 0.2985 | 0.0990 |
| | **0.01** | 1.0906 | 0.1976 | 0.3206 | 0.1055 |
| | $\sigma(\mu m)$ | $D_1$ $(\mu m^2/s)$ | $D_2$ $(\mu m^2/s)$ | $p_{12}$ | $p_{21}$ |
| SPT-2E | **0** | 1.0113 | 0.0992 | 0.2964 | 0.1016 |
| | **0.0001** | 0.9471 | 0.0983 | 0.2903 | 0.0972 |
| | **0.001** | 0.9782 | 0.0995 | 0.2967 | 0.0985 |
| | **0.01** | 0.9914 | 0.0983 | 0.3231 | 0.1060 |
| *Parameters used in the simulation* | Dynamical Error $R = \frac{1}{6}$ | *1* | *0.1* | *0.3* | *0.1* |
| | $\sigma(\mu m)$ | $D_1$ $(\mu m^2/s)$ | $D_2$ $(\mu m^2/s)$ | $p_{12}$ | $p_{21}$ |
| SPT-2 | **0** | 0.5718 | 0.0697 | 0.2314 | 0.0919 |
| | **0.0001** | 0.5483 | 0.0678 | 0.2226 | 0.0944 |
| | **0.001** | 0.5617 | 0.0698 | 0.2272 | 0.1017 |
| | **0.01** | 0.6969 | 0.1685 | 0.2466 | 0.0918 |
| | $\sigma(\mu m)$ | $D_1$ $(\mu m^2/s)$ | $D_2$ $(\mu m^2/s)$ | $p_{12}$ | $p_{21}$ |
| SPT-2E | **0** | 1.0087 | 0.1012 | 0.3049 | 0.0967 |
| | **0.0001** | 0.9582 | 0.0988 | 0.2930 | 0.0994 |
| | **0.001** | 0.9862 | 0.1001 | 0.3068 | 0.1072 |
| | **0.01** | 1.0066 | 0.0941 | 0.3163 | 0.1018 |

**Table 2.1:** Estimation of parameters using both SPT-2 and SPT-2E algorithms for simulated datasets with $\sigma = 0, 10^{-4}, 10^{-3}, 10^{-2}\ \mu m$ as the standard deviation of the static noise, $R = \frac{1}{6}$ for the dynamical error coefficient, and the following parameters set $\Theta = \{1\ \mu m^2/s, 0.1\ \mu m^2/s, 0.3, 0.1\}$.

| *Parameters used in the simulation* | **No dynamical noise** | *1* | *0.2* | *0.1* | *0.05* |
|---|---|---|---|---|---|
| | $\sigma(\mu m)$ | $D_1$ $(\mu m^2/s)$ | $D_2$ $(\mu m^2/s)$ | $p_{12}$ | $p_{21}$ |
| SPT-2 | **0** | 0.9936 | 0.1985 | 0.0923 | 0.0467 |
| | **0.001** | 1.0244 | 0.2013 | 0.0906 | 0.0487 |
| | **0.01** | 1.0121 | 0.2035 | 0.1021 | 0.0522 |
| | **0.1** | 1.3361 | 0.5039 | 0.1189 | 0.0599 |
| | $\sigma(\mu m)$ | $D_1$ $(\mu m^2/s)$ | $D_2$ $(\mu m^2/s)$ | $p_{12}$ | $p_{21}$ |
| SPT-2E | **0** | 0.9906 | 0.2002 | 0.0932 | 0.0464 |
| | **0.001** | 1.0207 | 0.2022 | 0.0907 | 0.0480 |
| | **0.01** | 1.0054 | 0.2020 | 0.1038 | 0.0519 |
| | **0.1** | 0.9916 | 0.1796 | 0.1217 | 0.0620 |
| *Parameters used in the simulation* | Dynamical Error $R = \frac{1}{6}$ | *1* | *0.2* | *0.1* | *0.05* |
| | $\sigma(\mu m)$ | $D_1$ $(\mu m^2/s)$ | $D_2$ $(\mu m^2/s)$ | $p_{12}$ | $p_{21}$ |
| SPT-2 | **0** | 0.6333 | 0.1355 | 0.0947 | 0.0491 |
| | **0.001** | 0.6671 | 0.1336 | 0.0881 | 0.0477 |
| | **0.01** | 0.6580 | 0.1385 | 0.0995 | 0.0483 |
| | **0.1** | 0.9515 | 0.4527 | 0.0902 | 0.0539 |
| | $\sigma(\mu m)$ | $D_1$ $(\mu m^2/s)$ | $D_2$ $(\mu m^2/s)$ | $p_{12}$ | $p_{21}$ |
| SPT-2E | **0** | 0.9605 | 0.1983 | 0.1038 | 0.0526 |
| | **0.001** | 1.0176 | 0.1964 | 0.0957 | 0.0496 |
| | **0.01** | 1.0011 | 0.1992 | 0.1086 | 0.0509 |
| | **0.1** | 0.9333 | 0.1879 | 0.0923 | 0.0520 |

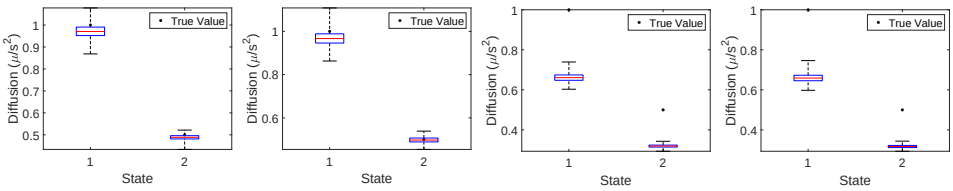
**Table 2.2:** Estimation of parameters using both SPT-2 and SPT-2E algorithms for simulated datasets with $\sigma = 0,\ 10^{-4}, 10^{-3},\ 10^{-2}\ \mu m$ as the standard deviation of the static noise, $R = \frac{1}{6}$ for the dynamical error coefficient, and the following parameters set $\Theta = \{1\ \mu m^2/s, 0.1\ \mu m^2/s, 0.3, 0.1\}$.

dynamical error, (b)on a dataset with standard deviation $\sigma = 0.0001\,\mu m$, (c) on a dataset with standard deviation $\sigma = 0.001\,\mu m$ for the static error and dynamical error with uniform illuminating profile $R = 1/6$, (d) on a dataset with standard deviation $\sigma = 0.0001\,\mu m$ for the static error and dynamical error with uniform illuminating profile $R = 1/6$. Figure 2.8 shows similar plots but using SPT-2 algorithm instead. The simulated datasets consist of 5000 frames sampled at 1 ms intervals.




(a) $\sigma$=$10^{-3}$, $R$=0. (b) $\sigma$=$10^{-4}$, $R$=0. (c) $\sigma$=$10^{-3}$, $R$=$\frac{1}{6}$. (d) $\sigma$=$10^{-4}$, $R$=$\frac{1}{6}$.

**Figure 2.7:** Estimated distributions of diffusion coefficients for each state of simulated 2-state particle tracks with 5000 steps sampled at 1ms intervals, and parameters $\Theta = \{1\,\mu m^2/s,\, 0.5\,\mu m^2/s,\, 0.1,\, 0.05\}$ using SPT-2E algorithm.




(a) $\sigma$=$10^{-3}$, $R$=0. (b) $\sigma$=$10^{-4}$, $R$=0. (c) $\sigma$=$10^{-3}$, $R$=$\frac{1}{6}$. (d) $\sigma$=$10^{-4}$, $R$=$\frac{1}{6}$.

**Figure 2.8:** Estimated distributions of diffusion coefficients for each state of simulated 2-state particle tracks with 5000 steps sampled at 1ms intervals, and parameters $\Theta = \{1\,\mu m^2/s,\, 0.5\,\mu m^2/s,\, 0.1,\, 0.05\}$ using the SPT-2 algorithm.

To study how the trajectory length affects our estimation, we simulated data with similar number of frames and same sampling time as the experimental data generated using organic fluorophores to label B cell surface receptors. We simulated 100 particles, each over 200 frames, with integration time equal to 0.0303 $s$, corresponding to 33 frames per second imaging.

Table 2.3 shows a comparison of the parameter estimation using our algorithm to fit two different types of trajectories: short ($\sim$ 200 frames) and long ($\sim$ 5000 frames). Additionally, a 2-fold difference between the diffusion coefficients is the minimum difference for a good performance of the SPT-2 algorithm. Here, we also conclude the same for our SPT-2E algorithm, as we can see on Table 2.3. For the shorter trajectories, the algorithm does not reliably recover the correct parameters when $\sigma = 0.1\ \mu m$. Here, there are two factors for the worse performance of the algorithm. First, the standard deviation of the slowest diffusion in this data is: $\sqrt{2 \times 0.5 \times 0.0303} = 0.17\ \mu m$, or roughly of the same order as the standard deviation of the noise. Moreover, the standard deviation of the fastest diffusion is $0.24\ \mu m$, which is approximately equal to the sum of the noise deviation and the slowest diffusion deviation. Therefore, it is not surprising that the algorithm cannot properly distinguish the noise from the slow state and the fast state.

Furthermore, we want to know how different would the estimations be if the input value of the static error in the algorithm did not correspond to the actual value on the simulation. Here, we use the same simulated data as in Table 2.2, but we set the assumed value of $\sigma$ to be 5 times larger than its true value. Table 2.4 shows the estimated parameters. The estimated parameters are close to their true values, however, the estimates are not as good as when the $\sigma_{Input}$ is the correct value for the standard deviation of the localization error. For small $\sigma$ ($0.001\ \mu m$), the recovered parameters are within 10% of the true value. Thus, it is important to have a accurate estimation of the static error, otherwise the estimates of the parameters of the model might not be accurate for large static errors ($\sigma$ =$0.01\ \mu m$).

## 2.4 Experimental Data Analysis

In this section, we analyze the datasets from previous work [1] using our SPT-2E algorithm. We compare two fluorescent labels commonly used for SPT of proteins. The first label is a small organic fluorophore (Cy3), and the second labelling method uses quantum dots (Qdots) as the fluorescent label. The advantage of quantum dots is that they do not photobleach,

| *Parameters used in the simulation* | **No dynamical noise** | *1* | *0.5* | *0.1* | *0.05* |
|---|---|---|---|---|---|
| 4 particles, 5000 frames, $\tau = 0.001s$. | $\boldsymbol{\sigma}$ $(\mu m)$ | $D_1$ $(\mu m^2/s)$ | $D_2$ $(\mu m^2/s)$ | $p_{12}$ | $p_{21}$ |
| | **0** | 0.9646 | 0.5030 | 0.0857 | 0.0511 |
| | **0.001** | 0.9695 | 0.4893 | 0.0973 | 0.0553 |
| | **0.01** | 1.0191 | 0.4891 | 0.1060 | 0.0558 |
| 100 particles, 200 frames, $\tau = 0.0303s$. | $\boldsymbol{\sigma}$ $(\mu m)$ | $D_1$ $(\mu m^2/s)$ | $D_2$ $(\mu m^2/s)$ | $p_{12}$ | $p_{21}$ |
| | **0** | 0.9822 | 0.4963 | 0.0920 | 0.0531 |
| | **0.01** | 0.9772 | 0.4938 | 0.0941 | 0.0559 |
| | **0.1** | 0.9266 | 0.4404 | 0.1014 | 0.0912 |
| *Parameters used in the simulation* | Dynamical Error $R = \frac{1}{6}$ | *1* | *0.5* | *0.1* | *0.05* |
| 4 particles, 5000 frames, $\tau = 0.001s$. | $\boldsymbol{\sigma}$ $(\mu m)$ | $D_1$ $(\mu m^2/s)$ | $D_2$ $(\mu m^2/s)$ | $p_{12}$ | $p_{21}$ |
| | **0** | 1.0055 | 0.4820 | 0.0958 | 0.0527 |
| | **0.001** | 1.0011 | 0.4814 | 0.0936 | 0.0525 |
| | **0.01** | 0.9836 | 0.4868 | 0.0810 | 0.0466 |
| 100 particles, 200 frames, $\tau = 0.0303s$. | $\sigma$ $(\mu m)$ | $D_1$ $(\mu m^2/s)$ | $D_2$ $(\mu m^2/s)$ | $p_{12}$ | $p_{21}$ |
| | **0** | 1.0290 | 0.4710 | 0.1360 | 0.0690 |
| | **0.01** | 0.9690 | 0.4510 | 0.1440 | 0.0930 |
| | **0.1** | 0.8770 | 0.000 | 0.2570 | 0.849 |

**Table 2.3:** Estimation of parameters using the SPT-2E model for simulated datasets with $\sigma = 0\mu m, 10^{-3}\mu m, 10^{-2}\mu m$ as the standard deviation of the static noise, $R = \frac{1}{6}$ for the dynamical error coefficient, and the following parameters set $\Theta = \{1\mu m^2/s, 0.5\mu m^2/s, 0.1, 0.05\}$.

allowing for longer trajectories. However, the size of quantum dots is of the same order as the tracked molecules. In [1], we sought to investigate if the two labelling methods yield similar results when using the same cells and receptors. One of the algorithms used to analyze the data was SPT-2. Here,

| **No dynamical noise** | $\sigma$ | $\sigma_{Input}$ | *1* | *0.2* | *0.1* | *0.05* |
|---|---|---|---|---|---|---|
| SPT-2E | $\mu$m | $\mu$m | $D_1$ $\mu m^2/s$ | $D_2$ $\mu m^2/s$ | $p_{12}$ | $p_{21}$ |
| | **0.001** | **0.005** | 1.0222 | 0.2078 | 0.0895 | 0.0461 |
| | **0.01** | **0.05** | 0.9287 | 0.1390 | 0.1033 | 0.0518 |
| Dynamical Error $R = \frac{1}{6}$ | $\sigma$ | $\sigma_{Input}$ | *1* | *0.2* | *0.1* | *0.05* |
| SPT-2E | $\mu$m | $\mu$m | $D_1$ $\mu m^2/s$ | $D_2$ $\mu m^2/s$ | $p_{12}$ | $p_{21}$ |
| | **0.001** | **0.005** | 0.9951 | 0.1929 | 0.0935 | 0.0495 |
| | **0.01** | **0.05** | 0.8509 | 0.1154 | 0.1034 | 0.0490 |

**Table 2.4:** Estimation of parameters using the SPT-2E model for simulated datasets with $\sigma = 0\ \mu m$, $10^{-3}\ \mu m$, $10^{-2}\ \mu m$ as the standard deviation of the static noise, $R = \frac{1}{6}$ for the dynamical error coefficient, and the following parameters set $\Theta = \{1\ \mu^2/s$, $\{0.2\ \mu^2/s, 0.1, 0.05\}$. In these table, the input standard deviation $\sigma_{input}$ is 5 times larger than the true value of $\sigma$. The estimates are not as close to the true values as when we use the correct value of $\sigma$, however, they are still in close proximity to the true parameters, demonstrating that the algorithm is quite robust to this mis-specification.

we further analyze our data, taking into account the errors, and compare with our previous results.

For our data, the illumination profile is uniform, because the shutter is open during the whole experiment. Therefore the dynamical coefficients $R_1$ and $R_2$ are $\frac{1}{3}$ and $\frac{1}{2}$ respectively. The localization accuracies were measured to be 0.036 $\mu m$ for Cy3 and 0.029 $\mu m$ for Qdot, by studying the apparent mobility of motionless stuck particles. The standard deviation of the displacements of stuck particles labelled by either Cy3 or Qdot is the localization accuracy [1].

Figures 2.9(a) and 2.10(a) are the equivalents of figures 3D and 3E in [1], which we reproduce (with permission) as Figure 2.9(c) and 2.10(c). There are a few differences between the results from the two algorithms. First, the

value of the slow diffusion coefficient is smaller when errors are corrected for. This is expected, because we saw that errors increase the apparent diffusivity of the slow state when they are not corrected for. Moreover, the slow diffusivity when errors are included is essentially zero, indicating that the molecules are probably not moving during the slow state. However, as we discussed before, the slow diffusion coefficient will approach zero if the standard deviation of the static error is larger than the characteristic diffusion distance of the slowest state. Therefore, we can conclude that the slow diffusivity for Qdot labelled receptors is smaller than 0.0139 $\mu m^2/s$, while for Cy3 labelled receptors it is smaller than 0.0214 $\mu m^2/s$. Both of these values are lower than values reported in [1] using the SPT-2 algorithm.

Furthermore, the diffusivity of the fast state for both labelling molecules is higher than the values estimated using the previous two-state algorithm. As shown in Table 2.1, the estimated values of the fast state diffusivity are smaller than the actual values when the SPT-2 is used to analyze data with errors. Thus, these higher values for the fast diffusivity are in agreement with our simulated data analysis above. The fast diffusivity is higher for Qdot-labelled proteins.

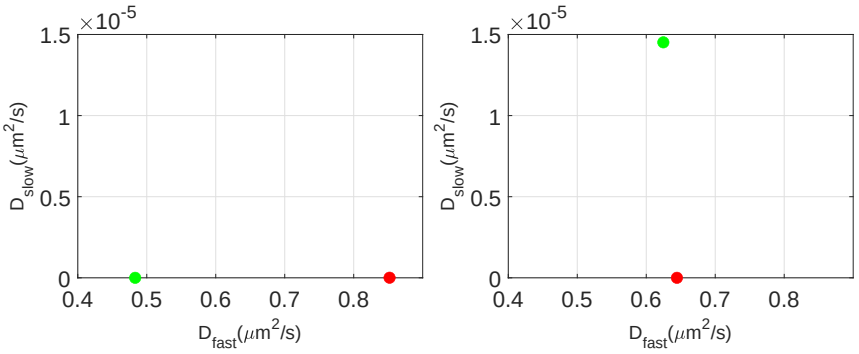
Figure 2.9(b) and 2.10(b) also provide the results for the experiment where latrunculin-A (LatA) was added to the sample. This substance disrupts the actin cytoskeleton of the cell. We found that the fast diffusion coefficient was similar for Qdot and Cy3-labeleld molecules, and greater than that observed for the control cells (for Cy3-labelled molecules). The actin cytoskeleton creates diffusion barriers limiting the area that an individual receptor can explore. Thus, when disrupted by LatA, there are less barriers limiting the motion of the receptors, which could explain the increase of diffusivity on the receptors.

In summary, for control experiments both labelling methods show a similar small mobility of the slow state, while the fast state of the Qdot experiment has a higher value compared to Cy3-labelled molecules. One might think that the movement of the Qdot-labelled molecule is not impaired by the labelling procedure as suggested by [1]. However, as we can see from the rate of transitions from slow (2) to fast (1) state and vice-versa, the rate

of transition to the slow state is approximately 2-fold higher for the Qdot labelling condition, compared to Cy3-labelled molecules in the control experiment. For the experiment with LatA, this difference is only approximately 1.3-fold higher when the receptors were labelled with Qdots. We conclude, as before, that Qdot-labelled molecules exhibit much lower mobility, on average, compared to molecules labelled with Cy3. One likely reason for this slower mobility in Qdot-labelled molecules is the close apposition of the cell membrane to the coverslip, resulting in Qdots causing frictional drag.

When fitting experimental data to this model, we forced the data to fit a two-state model, even if the data would be better explained by a 3 state model, for example. One of the consequences is that the diffusion coefficients for a 3-state model would be averaged resulting in two coefficients. The displacements whose underlying diffusion coefficient is the third state would be divided among the other two states by the algorithm, resulting in the algorithm estimating a larger diffusion coefficient for the slow state, and a smaller diffusion coefficient for the fast state compared with the values of the 3-state model. The three state model is used here as an example of possible implications of forcing the data to a 2-state model. This could be one possible interpretation of the above results. Because Cy3-labelled molecules are smaller, they may enter many different regions of the cellular membrane, including very crowded regions, whereas Qdot-labelled molecules may not enter some regions due to their sizes, and might have a higher chance of being stuck in a very crowded area as well. We will investigate multi-state model in Chapters 3 and 4.

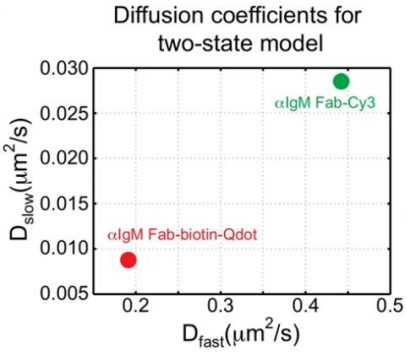
In addition, fewer diffusion barriers could explain the reason for the increase of the slow diffusivity for both Cy3 and Qdot-labelled molecules. The fast diffusivity decreased for Qdot-labelled molecules, and increased for Cy3-labelled molecules. The re-organization of the cellular membrane might creates a varied of regions with different levels of crowding, size, and mobility. These should differently affect the labelled molecules, given the difference in size of the labels.




(a) Control.




(b) LatA.




(c)Taken from Abraham et al. [1] with permissions.

**Figure 2.9:** Diffusivity estimates for SPT-2E Cy3-labelled trajectories for a control and LatA experiment condition, and Qdot-labelled trajectories on a Control and LatA experiment condition.(a) Diffusion coefficient of slow state (y-axis) vs Diffusion coefficient of fast state (x-axis), red for Qdot-labelled proteins and green for Cy3-labelled proteins. (b) Same as (a) but for experiments where LatA is added on the cell. Diffusivities are larger for LatA experiments as expected, since the actin cytoskeleton is disrupted.(c) Graph showing the diffusion coefficients using SPT-2 taken from Abraham et al. [1] with permission under the Creative Commons License. To view a copy of the license visit: http://creativecommons.org/licenses/by/4.0/.

## 2.5 State Segmentation: Adapted Forward-Backward Algorithm

To further understand the heterogeneity of the trajectories, we can perform a state segmentation. The state segmentation is the estimation of which state is most probable at each time point of the data. This analysis can potentially reveal spatially heterogeneous phenomena. The Forward-Backward algorithm is a dynamical programming algorithm that finds the most likely sequence of hidden states $s_1$, $s_2$, ..., $s_N$ for the sequence of displacements $\Delta_1$, ..., $\Delta_N$.

First, let $\beta_j$ be the backward variable. Similar to the forward variable $\alpha_j$, $\beta_j$ is the probability of observing the future sequence of displacement $\Delta_{j+1}$, ..., $\Delta_N$, given that $\Delta_j$ is in state $s_j$. Therefore,

$$\beta_j(i) = P\left[\Delta_{j+1} \cdot \Delta_{j+2} \cdots \Delta_N, s_j = i|\Theta\right]. \tag{2.17}$$

Because we have a tridiagonal covariance matrix, we needed to make some changes in the calculation of $\beta_j$. In this adapted forward-backward algorithm, we calculated $\beta$ at every two steps, yielding a probability distribution of pairs of displacements. In this way, we include the off-diagonal terms and sum over all possible previous and current states. This step is very similar to the calculation of $\alpha$ in the adapted forward method. Then, we have:

$$\beta_j(i) = \sum_{s_{j+1}=1}^{2} \sum_{k=1}^{2} \beta_{j+2}(k) p_{s_j s_{j+1}} p_{s_{j+1} k} P\left(\Delta_{j+1}\Delta_{j+2}|s_{j+1} = m, s_{j+2} = k, \Theta\right), \tag{2.18}$$

where $P\left(\Delta_{j+1}|s_{j+1} = k, \Theta\right)$ is given by Equation 2.16.

Finally, the probability of being in state $i$ at time step $j$ for an observed track $\mathbf{\Delta}$, conditional on $\Theta$ is proportional to

$$P\left(s_j = i|\mathbf{\Delta}, \Theta\right) = \alpha_j(i)\beta_j(i). \tag{2.19}$$

Using EquationEquation 2.19 we can infer the most probable sequence of states for a track $\mathbf{\Delta}$.

We tested the modified forward-backward algorithm using the simulated data and inferred parameters' values described in Table 2.2. We compare the estimated state for each displacement with the true state for each displacement to calculate the percentage correctness of the sequencing, i.e.

$$\text{Correctness} = \frac{\text{Number of correct estimated states}}{\text{Total number of displacements}}.$$

Table 2.5 shows this quantity calculated using simulated data. We achieve a good accuracy for the sequencing of states, with over 80% of states classified correctly even for the trajectories with highest errors. However, perhaps surprisingly, the two methods perform about equally well. This result is explained as follows.

| Errors | % SPT-2E | % SPT-2 |
|---|---|---|
| 0 | 89.67 | 90.46 |
| 0.001 | 90.25 | 91.59 |
| 0.01 | 89.92 | 91.15 |
| 0.1 | 82.06 | 81.00 |

**Table 2.5:** Correctness of sequencing of states for simulated data with both dynamical and static errors, using the SPT-2E model, and the SPT-2 model. Fits are to simulated data with continuous intensity profile function ($R_1 = \frac{1}{3}$ and $R_2 = \frac{1}{2}$). Different values were assumed for the standard deviation of static noise, $\sigma = 0$ $\mu m$, 0.001 $\mu m$, 0.01 $\mu m$, 0.1 $\mu m$.

The diffusion coefficient estimates are the main difference between the two HMM models. The SPT-2 sequencing algorithm assumes that the displacements are independents, whereas the SPT-2E algorithm takes into account the correlations caused by experimental errors in positions. In the end, the different assumptions of the algorithms are balanced out by the different diffusivity estimates in such a way that both sequencing algorithms arrive at similar state classifications. Note that the Viterbi algorithm, which calculates the probability of the most likely partial sequence of states and uses it

to segment the data, generally performs better than the forward-backward algorithm.

Given that the accuracy on the results are similar for both sequencing methods, we expect the sequencing of the experimental data to be similar as the one on [1]. Table 2.6 shows the percentage of slow states in the state sequences using the SPT-2E for the experimental data from [1]. We obtain similar results to those presented in [1], with a slight increase in the values for the percentage on slow state for all experiments. However, the basic conclusions are the same: the percentage of Qdot-labelled molecules in the slow state are higher than Cy3-labelled, suggesting lower mobility of Qdot-labelled molecules. Moreover, upon addition of LatA, Cy3-labelled molecules became more mobile, and had a smaller fraction of slow states compared to the control. In contrast, the difference betweem Qdot-labelled experiments with LatA, and the corresponding Qdot control experiments is only 3%. This suggests that Qdot-labelling underestimates the effects of LatA on the molecular motion.

| Experiment | Cy3 | Qdot |
|---|---|---|
| Control | 82.37 | 93.44 |
| LatA | 66.31 | 90.44 |

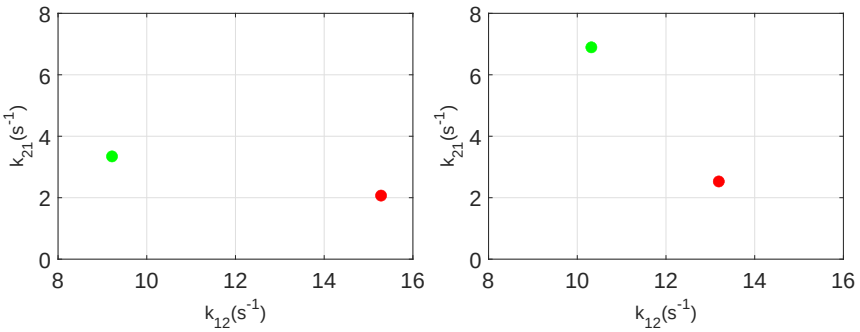
**Table 2.6:** Percentage of slow states obtained using the SPT-2E algorithm for the experimental data from [1]. Two different experimental conditions were performed using Cy3-labelled molecules and Qdot-labelled molecules: a control, and adding LatA to the sample.

## Summary

We wanted to understand how the lack of precision in experimental position measurements affects the parameter estimation for a multi-state diffusion process. To address this question, we developed an algorithm based on the model of Das et al. [24], and Berglund's description of the measurement process.

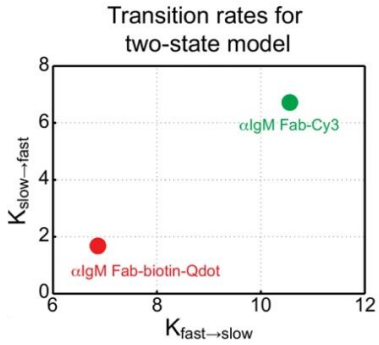
Using simulated data, we first validated our SPT-2E algorithm, and checked its robustness. After that, we applied our SPT-2E algorithm to experimental data from [1]. Finally, we compared the results from our SPT-2E algorithm with the results from Abraham et al. [1].

We found that the SPT-2E algorithm estimates the parameters set $\Theta$ with good accuracy. For simulated data including only static errors, both the SPT-2 and SPT-2E algorithms recover the parameters, except when $\sigma$ is large in which case SPT-2 overestimates the diffusion coefficients. For simulated data including errors, SPT-2 fails to recover any of the parameters accurately, whereas SPT-2E successfully recovers all parameters. Overall, we find that the SPT-2E algorithm is a significant advance over the SPT-2 algorithm for experimental data with appreciable localization errors - for example as when tracking cell surface receptors labelled with fluorescent proteins.




(a) Control.

(b) LatA.




(c)Taken from Abraham et al. [1] with permissions.

**Figure 2.10:** Results of SPT-2E for Cy3-labelled trajectories for a control and LatA experiment condition, and Qdot-labelled trajectories on a Control and LatA experiment condition. (a) Kinetic rates from slow to fast $k_{21}$, and fast to slow $k_{12}$. Kinetic rates are approximated by $k_{12}\frac{\sim p_{12}}{\tau}$, where $\tau$ is the time duration of a frame. Green is for Cy3-labelled trajectories, and red for Qdot-labelled trajectories. (b) Same as (a) but for experiments where LatA is added to the cell. Qdot-labelled transition rate to slow states is around 6.5 times larger than to fast states, and 4 times larger than Cy3 transition rate to slow states. (c) Graph showing the kinetic rates using SPT-2 taken from Abraham et al. [1] with permission under the Creative Commons License. To view a copy of the license visit: http://creativecommons.org/licenses/by/4.0/.

# CHAPTER 3

## Infinite State Hidden Markov Model

Advances in fluorescence microscopy imaging have allowed us to develop an increasingly well-resolved picture of the spatial distribution and spatiotemporal mobility of important cellular proteins. Our rapidly developing knowledge has supported the development of quantitative theories of intercellular communication, cell surface receptor signaling and downstream responses.

Single particle tracking (SPT) is a technique that is of particular importance in defining the modes of protein mobility on the cell surface and within the cell [117]. Typically, cellular proteins are specifically labelled with a fluorescent tag. By labelling only a small fraction of molecules, individual fluorescent tags can be localized in a series of images and image analysis software can be used to link tag positions and thus obtain particle trajectories [18]. Once we have the trajectories, we can analyze their properties. Assuming that the particles are only subject to thermal noise, it is natural to analyze the tracks as representatives of simple Brownian motion and estimate the diffusion coefficient [31, 99, 112], while taking into account limitations in particle localization accuracy [6, 79, 101, 135].

Particle trajectories extracted from cellular proteins have often exhibit

deviations from simple diffusive behaviour, which has been attributed to transient binding to cellular structures, transient confinement in subcellular domains, directed motion under the influence of molecular motors, anomalous diffusion, etc [19, 28, 75, 113, 120, 126]. In order to analyze transient behaviour within individual tracks, dynamic multi-state models have been developed and analyzed over the last ten years [9, 13, 24, 69, 82, 136, 142]. These methods explicitly assume that particles can switch among a specific number of diffusive states. For example, Das et al. used a two-state Hidden Markov Model (HMM) to fit particle tracking data for the surface receptor LFA-1 on the surface of T cells [13, 24]. In this model, particles were allowed to transition between two diffusive states, each characterized by a particular diffusion constant. The diffusion constants and transition rates between the states were fit to the data. At the time, previous experimental evidence pointed to a large reduction in LFA-1 mobility upon binding with cytoskeletal components, suggesting that the model could be a reasonable approximation of the dynamics of LFA-1 as it transitioned between bound and unbound states. More refined HMM approaches have subsequently been developed, reflecting finite localization errors and including particle capture within potential wells [7, 93, 121, 122].

However, in the more general setting, how can one *a priori* ascertain the best number of states to explain the data and provide insights into the biology of the tracked particles? Besides being a biological question of great interest, the number of possible states changes the statistical model and hence, the number of parameters to estimate. Therefore, estimating the number of states is an important problem in multi-state SPT analysis.

One solution to this problem was recently introduced by Linden and Elf [67], where a variational Bayesian approximation was used to select the best model and a predictive approach was implemented using cross-validation. An alternative approach was previously described by Koo et al. [61], where ensembles of short particle trajectories were analyzed through an expectation-maximization approach to a Gaussian mixture model, yielding the number of states, their diffusivities, and the stationary probabilities that particles inhabit each state. Here, we will attack the problem using

a Bayesian nonparametric approach that allows the parameter space to be infinite-dimensional. The so-called Infinite Hidden Markov Model (iHMM) is a nonparametric model [35] that has recently been applied to FRET data by Sgouralis and coworkers [115, 116] to estimate the number of conformations of a molecule and simultaneously infer kinetic parameters for each conformational state. We use these concepts to develop a novel tool to analyze single particle tracking data under the assumption that the trajectories follow a Markov chain, where each element of the chain is a diffusion process. We seek to infer the number of diffusive states, the transition rates among states and the diffusion coefficient defining each state from the available data.

In this chapter, we will begin by specifying the HMM model that we wish to apply to SPT data, and then follow the thorough exposition of Sgouralis et al. [115] to generalize to the infinite-dimensional (iHMM) setting. We validate the model using simulated data and provide a technical improvement to the algorithm that assists with convergence for the problem at hand. Finally, we apply our method to real data from experiments using TIRF microscopy to visualize motion of surface receptors on the membrane of live B cells[1] and discuss biological implications and possible future directions for study.

## 3.1 The Dirichlet Process

The Dirichlet process, together with a Markov model, are the main components of the infinite hidden Markov model. This Dirichlet process is an infinite-dimensional generalization of a Dirichlet distribution. The construction paves the way to define a Markov chain over an infinite state space, creating a framework for multimodel inference [5, 125, 130, 131]. Here, we start with an overview of the Dirichlet distribution and then generalize it to the infinite dimensional domain.

### 3.1.1 Dirichlet Distributions

The Dirichlet distribution can be thought of a distribution over distributions, meaning that one samples a probability mass function [35, 38, 124, 130]. An example of a probability mass function is a die. A die with 6 faces defines a probability of mass function of size 6, $\pi = (\pi_1, \pi_2, \ldots, \pi_6)$, and each $\pi_i$, gives the probability of rolling face $i$ of the die. Now, imagine we have a bag with 1000 dice, and each die is manufactured in a different way, so that the probability of rolling a face is different among the dice. Therefore, each die has a different probability mass function. Every time one draws a die from this bag, one draws a random probability mass function $\boldsymbol{\pi}_j$, where $0 < j \leq 1000$. The Dirichlet distribution models the randomness of this experiment - drawing a die from a bag of dice - drawing a probability mass function.

Suppose $Z \sim \text{Gamma}(\alpha, \beta)$ is distributed as a Gamma distribution with shape $\alpha > 0$ and rate $\beta > 0$. This distribution is defined by

$$p(Z \mid \alpha, \beta) = \frac{\beta^{\alpha} Z^{\alpha-1} \exp(-\beta x)}{\Gamma(\alpha)}. \tag{3.1}$$

The Dirichlet distribution can be defined from a collection of such Gamma random variables [35]. Let $Z_i$, for $0 < i \leq k$, be independent random variables following a Gamma distribution $\Gamma(\alpha_i, 1)$ with shape equal to $\alpha_i$ and rate equal to 1. Define $\pi_i = \frac{Z_i}{\sum_{j=1}^{k} Z_j}$, for $0 < i \leq k$. The Dirichlet distribution is defined as the distribution of $(\pi_1, \ldots, \pi_k)$ with parameters $(\alpha_1, \cdots, \alpha_k)$, i.e.,

$$(\pi_1, \cdots, \pi_m) \sim \text{Dir}\,(\alpha_1, \cdots, \alpha_k)\,.$$

$\pi_i$ is always larger than 0 and smaller than 1. Moreover, $\sum_i \pi_i = 1$. Therefore, the domain of the Dirichlet distribution lies over the $(k-1)$-dimensional probability simplex. The density is given by:

$$p(\pi_1, \cdots, \pi_k \mid \alpha_1, \cdots, \alpha_k) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_k)}{\Gamma(\alpha_1) \cdot \vdots \cdot \Gamma(\alpha_k)} \prod_1^{k-1} \pi_i^{\alpha_i} \left(1 - \sum_1^{k-1} \pi_i\right)^{\alpha_k}, \tag{3.2}$$

where $\pi_k = 1 - \sum_1^{k-1} \pi_j$. (3.2) can be calculated by substitution of equation (3.1) into:

$$p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) = \prod_1^k p(\pi_i \mid \alpha_i) = \frac{\prod_1^k p(Z_i \mid \alpha_i)}{p(\sum_1^k Z_i \mid \alpha_i)}.$$

Given that $\boldsymbol{\pi}$ itself is a distribution, the Dirichlet distribution is called the distribution of distributions [38, 124].

To better understand how the parameter $\boldsymbol{\alpha}$ changes the shape of the distribution, we plot the Dirichlet distribution for $k = 2$, and 3, and different values of $\alpha$. Thus, we have a probability mass function on a 1-dimensional probability simplex, and on a 2-dimensional probability simplex, respectively.

Note that for $k = 2$, the Dirichlet distribution become a Beta distribution,

$$p(\boldsymbol{\pi}; \boldsymbol{\alpha}) = \frac{\Gamma(\alpha_1 + \alpha_2)\pi_1^{\alpha_1 - 1}(1 - \pi_1)^{\alpha_2 - 1}}{\Gamma(\alpha_1)\Gamma(\alpha_2)}.$$

Figure 3.1 shows the distribution for the case $K = 2$, i.e. the Beta distribution. When both values of $\alpha$ are the same, we have a symmetric curve. For the case where $\alpha_1 = \alpha_2 = 1$, we have a uniform distribution. For $\alpha_1 = \alpha_2 < 1$, we have higher values on the boundaries. For $\alpha_1 = \alpha_2 > 1$, the peak is on 0.5. When $\boldsymbol{\alpha}$ has different values, the curve has higher values on small $\pi_1$ for $\alpha_1 < 1$, and for $\alpha_1 > 1$ the higher values are on large $\pi_1$.

For $k = 3$, we are dealing with a 2-dimensional probability simplex. The graphs plotted in Figure 3.2 show how the density changes for different $\alpha$. For $\alpha = [1, 1, 1]$, we have an uniform distribution over the domain. When $\alpha = [6, 6, 6]$, the density peak is on the middle of the space. For $\alpha = [0.1, 0.1, 0.1]$, the highest values are on the boundary of the domain. When

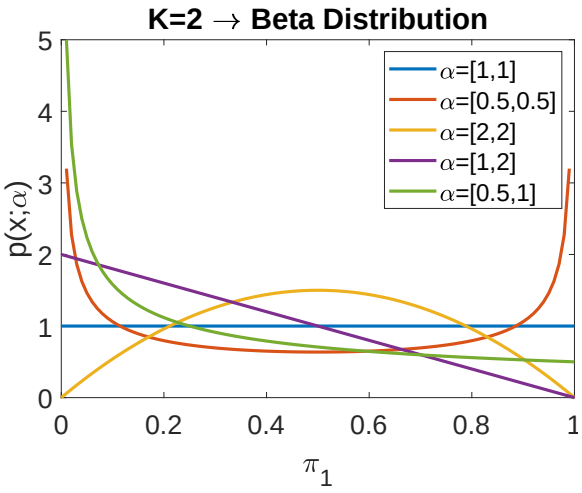



**Figure 3.1:** Dirichlet Distribution for $k = 2$, where the probability mass function is on a 1-dimensional probability simplex. Colors represent different values for $\boldsymbol{\alpha}$.

the elements of $\boldsymbol{\alpha}$ are different, the density is not symmetric any more, and the peak moves in the direction of the largest $\alpha_i$.

We can separate the $\boldsymbol{\alpha}$ values using two features: the absolute value of each element, and the relative value among the elements. First, their absolute value indicates how much belief we have on the prior, so for higher values of $\boldsymbol{\alpha}$, the sampled distribution is closer to the prior [124]. By "prior", we mean the relative ratio among the elements of $\alpha$. For the example of a die with 6 faces, $\boldsymbol{\alpha}$ has 6 elements. From our prior knowledge we expect that the probability of drawing any face to be the same, and equal to $\frac{1}{6}$. Then, our prior is $(\frac{1}{6}, \ldots, \frac{1}{6})$. Therefore, the elements of $\boldsymbol{\alpha}$ are equal, and their ratios are the same as the prior ratios, since we do not want to add any bias to any face of the dice. Figure 3.2 can be a representation of a die with 3 faces. For $\boldsymbol{\alpha} = (1, 1, 1)$, all the faces have the same chance to be drawn, however our belief in this prior is neutral. We are not sure whether this die is not biased. Therefore, we have a uniform distribution for the Dirichlet process. If we are more confident that our dice is not biased, we increase the values $\boldsymbol{\alpha}$ to 6, and there we have higher probability around $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ distribution.

If we decrease $\boldsymbol{\alpha}$ values, we know that the prior is biased, and the higher values are on the edges. If the die is biased, and we know which faces have higher chances, we can set $\boldsymbol{\alpha}$ with higher values for these faces.

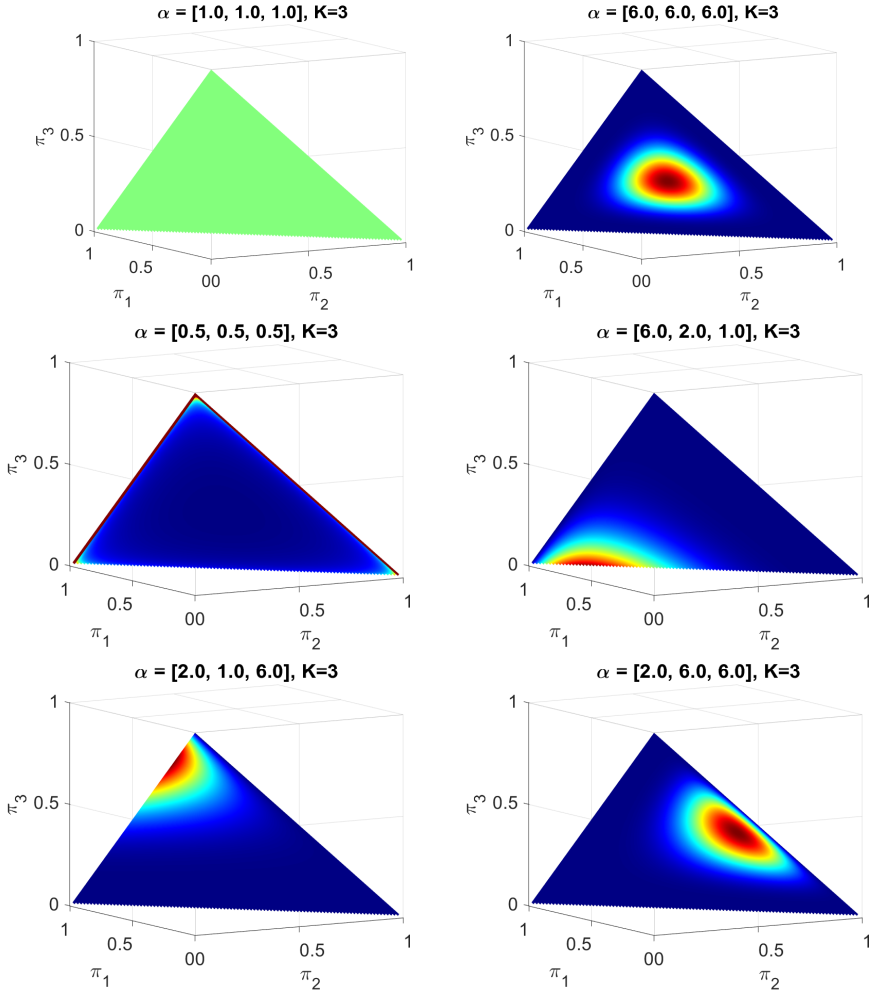



**Figure 3.2:** Illustrating the $K = 3$ Dirichlet distribution for different values of the input vector $\alpha$.

### 3.1.2 Sampling from a Dirichlet Distribution

There are different ways that one can generate samples of a Dirichlet distribution [38]. Here, we explain one of them known as the stick-breaking construction. The stick-breaking process constitutes of the following steps:

1. First, draw $a_1 \sim \text{Beta}\left(\alpha_1, \sum_{i=2}^{k} \alpha_i\right)$. Set $\beta_1 = a_1$.
2. For $1 < j \leq k-1$, draw $a_j \sim \text{Beta}\left(\alpha_j, \sum_{i=1, i\neq j}^{k} \alpha_i\right)$. Then, set $\beta_j = a_j \prod_{i=1}^{j-1} (1 - a_i)$.
3. $\beta_k$ is equal to the length of the remaining stick.

The idea behind this process is as follows: Imagine that one has a stick of unit length, which is then broken at a point $a_1$ drawn from a Beta distribution, i.e. $\text{Beta}\left(\alpha_1, \sum_{i=2}^{k} \alpha_k\right)$. Then using the rest of the stick of length $(1 - a_1)$, select a new break-point $\beta_2 = a_2(1 - a_1)$, where $a_2 \sim \text{Beta}\left(\alpha_2, \sum_{i\neq 2, i=1}^{k} \alpha_k\right)$. Continuing this sequence of tasks, one ends up with variables $a_i$ that have the two required probabilistic properties: they are each less than one, while their summation is equal to one, the length of the original stick. Moreover, they are distributed accordingly a Dirichlet distribution with parameter $\boldsymbol{\alpha}$, $\text{Dir}(\boldsymbol{\alpha})$, as we shown in the subsection 3.1.3.

Another approach for sampling a Dirichlet distribution is using a Gamma distribution, following the same idea as explained in the definition of Dirichlet distribution given above. First, $Z_i$'s are drawn from a Gamma distribution for each component of the Dirichlet distribution. Next, we renormalize $Z_i$'s by dividing it by the sum over all $Z_i$. Thus, $\beta_i = \frac{Z_i}{\sum_i Z_i} \leq 1$, and $\sum_i \beta_i = 1$.

### 3.1.3 Properties of Dirichlet distributions

#### Neutrality

One important property of Dirichlet distribution is neutrality. A random vector defined such that its elements add up to a fixed value is said to be neutral, when every element is independent from the vector made

by the other elements as proportions of their total. To be more precise, let $\boldsymbol{A} = (A_1, \dots A_j, \dots A_m)$, and $\sum_1^m A_i = 1$. Upon removing $A_j$, the vector of remaining elements is $\boldsymbol{R_A} = (A_1, \dots, A_{j-1}, A_{j+1}, \dots, A_m)$. We say that $A_j$ is neutral if the joint probability $P\left(A_j, \frac{\boldsymbol{R_A}}{1-A_j}\right)$ is equal to $P\left(A_j, \frac{\boldsymbol{R_A}}{1-A_j}\right) = P\left(A_j\right) P\left(\frac{\boldsymbol{R_A}}{1-A_j}\right)$.

For a random vector following a Dirichlet distribution, each one of the elements is neutral. Therefore the vector is said to be completely neutral. The demonstration of this property is given in the Appendix and follows the technical report by Frigik et al.[38]. This demonstration consists of using the change of variable formula to rewrite the joint density $P\left(A_j, \frac{\boldsymbol{R_A}}{1-A_j}\right)$ as a multiplication of two independent densities for every element of $\boldsymbol{A}$. The final expression is given by:

$$p\left(A_j, \frac{\boldsymbol{R_A}}{1-A_j}\right) \mid \boldsymbol{\alpha}) = \left(\frac{\Gamma(\sum\limits_{\substack{i=1\\ i\neq j}}^{k} \alpha_i)}{\prod\limits_{\substack{i=1\\ i\neq j}}^{m} \Gamma(\alpha_i)} \left(\prod_{\substack{i=1\\ i\neq j}}^{m} Y_i^{\alpha_i - 1}\right)\right)\left(\frac{\Gamma(\sum_i^k \alpha_i)}{\Gamma(\alpha_j)\Gamma(\sum\limits_{\substack{i=1\\ i\neq j}}^{k} \alpha_i)} Y_j^{\alpha_j - 1}(1 - Y_j)^{\sum\limits_{i=1, i\neq j}^{m} \alpha_i - 1}\right), \tag{3.3}$$

where $Y_i = \frac{A_1}{1-A_j}$ for $i = 1, \dots, j-2, j+1, \dots, m$, $Y_{j-1} = 1 - \sum\limits_{\substack{i=1\\ i\neq j-1, j}}^{m} A_i$, and $Y_j = A_j$. The first term is a Dirichlet distribution and represents the joint density of $\boldsymbol{Y}_{-j}$, or $(Y_1, \dots, Y_i, \dots, Y_m)$ for $0 < i \leq m$, and $i \neq j$, conditioned to $Y_j$. The second term is a Beta distribution and expresses the density of

$Y_j$. Therefore, equation (3.3) implies that:

$$(\boldsymbol{Y}_{-j} \mid Y_j) \sim \mathrm{Dir}(\boldsymbol{\alpha}_{-j}), \tag{3.4}$$

$$\left(\frac{\boldsymbol{A}_{-j}}{1 - A_j} \mid A_j\right) \sim \mathrm{Dir}(\boldsymbol{\alpha}_{-j}), \tag{3.5}$$

$$(\boldsymbol{A_{-j}} \mid A_j) \sim (1 - A_j)\,\mathrm{Dir}(\boldsymbol{\alpha_j}) \tag{3.6}$$

From the equations (3.3), and (3.6) one can rigorously prove that the stick-break construction generates samples of a Dirichlet distribution [38]. Here, we give a brief summary of the proof.

The first step of the stick-breaking construction consists of drawing $A_1 \sim \mathrm{Beta}\left(\alpha_1, \sum_{i=2}^{m} \alpha_i\right)$, and set $\pi_1 = A_1$. Next, we need to sample $A_2, \ldots, A_m$ given $A_1$. From equation (3.6), we know that $\boldsymbol{A_{-1}}$ is distributed as $(1 - A_1)\mathrm{Dir}(\boldsymbol{\alpha_{-j}})$. $1 - A_1$ is the length of the remaining stick that we will break in pieces according to a vector of proportions distributed as a Dirichlet distribution. If we do this recursively, for $A_2, \ldots, A_{m-2}$, we obtain that $(A_{m-1}, A_m \mid (A_1, \cdots, A_{m-2})) \sim \prod_{i=1}^{m-2} (1 - A_i)\mathrm{Dir}(\alpha_{m-1}, \alpha_m)$. Since we have that $\sum A_i = 1$, then we only need to split the remainder of the stick into two pieces by drawing $A_{m-1} \sim \prod_{i=1}^{m-2} (1 - A_i)\mathrm{Beta}(\alpha_{m-1}, \alpha_m)$. Then, if in every iteration we draw a break-point of the stick from a Beta distribution, and rescale the stick by its remainder, we have a random vector whose elements are Dirichlet distributed, and defined over a probability simplex [38].

**Dirichlet Distribution as a Conjugate Prior** Another relevant property of the Dirichlet distribution is that they are conjugate to the Multinomial distribution (or Categorical distribution) [38, 124]. The Multinomial distribution describes the probability that an event $i$ occurs $x_i$ times, given that the probability of an event $i$ to happen is given by $q_i$. The probability mass function is:

$$p((x_1, x_2, \cdots, x_k) \mid, (q_1, q_2, \cdots, q_k)) = \frac{n!}{x_1! x_2! \cdots x_k!} \prod_{i=1}^{k} q_i^{x_i} \tag{3.7}$$

Bayes rule gives the relation between the posterior distribution with the likelihood of the data, and the prior distribution of the parameter. The Bayes' theorem is stated as: $p(q \mid \text{Data}) = \frac{p(\text{Data}|q)p(q)}{p(\text{Data})}$, where $p(q)$ is the prior distribution of the unknown (parameter), $p(\text{Data})$ is the marginal likelihood (evidence) and $p(q \mid \text{Data})$ is the posterior distribution. Since the evidence does not depend on the parameters, we have $p(q \mid \text{Data}) \propto p(\text{Data} \mid q)p(q)$. A distribution is said to be a conjugate prior for the likelihood distribution, when the posterior distribution has the same shape as the prior distribution.

Applying equation (3.7) and the Dirichlet distribution as likelihood and prior respectively on Bayes rule, we obtain:

$$p(q \mid x) \propto \frac{n!}{x_1!x_2!\cdots x_k!} \prod_{i=1}^{k} q_i^{x_i} \frac{\Gamma(\alpha_1 + \alpha_2 + \cdots + \alpha_k)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} q_i^{\alpha_i - 1} \tag{3.8}$$

$$p(q \mid x) \propto z \prod_{i=1}^{k} q_i^{x_i + \alpha_i - 1} \tag{3.9}$$

$$p(q \mid x) \propto Dir(\alpha + x), \tag{3.10}$$

where $x$ is the observed data.

Since the posterior also follows a Dirichlet distribution when the likelihood is a Multinomial distribution, then the Dirichlet distribution is a conjugate prior for the Multinomial distribution. This facilitates Bayesian inference, since we have a closed formula for the posterior.

**Aggregation**

Another important property of the Dirichlet distribution is the aggregation property. It follows from the additive property of the Gamma distribution [35]. The additive property says that if $Z_1 \sim \text{Gamma}(\alpha_1, 1)$, $Z_2 \sim \text{Gamma}(\alpha_2, 1)$, and $Z_1$ and $Z_2$ are independents, then $Z_1 + Z_2 \sim \text{Gamma}(\alpha_1 + \alpha_2, 1)$ [35].

Let $\boldsymbol{Y}$ be distributed by a Dirichlet distribution with parameters $\boldsymbol{\alpha}$,

$$(Y_1, Y_2, \ldots, Y_N) \sim \mathrm{Dir}\,(\alpha_1, \alpha_2, \ldots, \alpha_N)\,.$$

The aggregation property allow us to put parts of the vector $\boldsymbol{Y}$ together, and the final vector is still distributed as a Dirichlet distribution with the following parameters,

$$\left(\sum_{i=1}^{a_1} Y_i,\ \sum_{i=a_1+1}^{a_2} Y_i, \cdots,\ \sum_{i=a_{n-1}+1}^{N} Y_i\right) \sim \mathrm{Dir}\left(\sum_{i=1}^{a_1} \alpha_i,\ \sum_{i=a_1+1}^{a_2} \alpha_i, \cdots,\ \sum_{i=a_{n-1}+1}^{N} \alpha_i\right), \tag{3.11}$$

where $a_1, \ldots, a_{n-1}$ are integers and $1 \leq a_i < N$, and $0 \leq n \leq N$ [35].

Thus, we can transform a distribution sampled from a Dirichlet distribution by adding its components in groups. This transformed distribution still follows a Dirichlet distribution, and their parameters are also added together [35].

The Dirichlet process, $\mathrm{DP}(\alpha, \boldsymbol{\beta})$, is a stochastic process whose range is a set of probability distributions [130]. It is known as a generalization of the Dirichlet distribution to the infinite dimensional domain. The analogy to the dice bag still applies, however one realization of a Dirichlet process is a drawn from a bag of dice with an infinite number of faces.

The bag of dice is the range of the Dirichlet Process, where each die is a distribution probability itself. Then, one way to know each die's number of faces is to draw a die from the bag, and then throw the die many times. Every time we throw, we can either get a face that has already being drawn or get a new face. The probability distribution of the die is over the faces that already came, and the possible other faces that did not appear. Therefore, the Dirichlet Process allows the dimension of the space grows as we throw the dice and acquires new faces, i.e. the dimension grows with the data. The probability distributions of a DP are defined over a finite-dimensional

simplex, and the size of the simplex grows with the data. For the dice example, the simplex size is the faces that were sampled on the experiments and the complement of the set of these faces to integers. One can think of these sets as partitions of the integers.

Let $\beta$ be a probability distribution over $\Theta$. $\Theta$ is the state space of the probability distribution. It can be a vector space, the set of real numbers, the integers, and so on. This space defines what we are sampling. In other words, it is the set defining the random variable of $\beta$. For the dice example, $\Theta$ is the integers. Then, one can say that $P$ is a $\mathrm{DP}(\alpha, \beta)$ with concentration parameter $\alpha$ and base distribution $\beta$ if $(P(A_1), \ldots, P(A_n)) \sim \mathrm{Dir}\left(\alpha\beta(A_1), \ldots, \alpha\beta(A_n)\right)$ for every finite partition of $\Theta$ [35, 124, 130]. The stick-breaking construction representation of a Dirichlet process is thus given as follows [124]:

$$a_k|\alpha, \beta \sim \mathrm{Beta}(1, \alpha), \tag{3.12}$$

$$\phi_k|\alpha, \beta \sim \beta, \tag{3.13}$$

$$\pi_k = a_k \prod_{1}^{k-1} \left(1 - a_i\right), \tag{3.14}$$

$$P(A) = \sum_{i=1}^{\infty} \pi_i \delta_{\phi_i}(A) \tag{3.15}$$

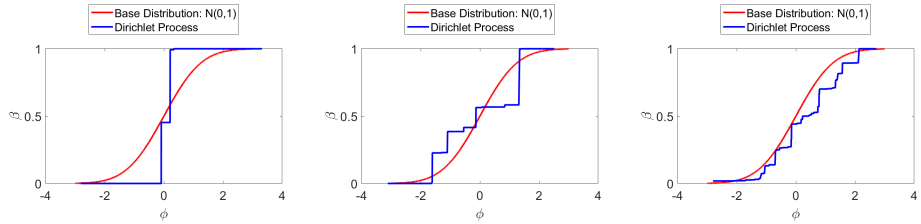
where $A$ is a set of the infinite sample space, $\delta_{\phi_i}(A)$ is the Dirac function, so $\delta_{\phi_i}(A) = 1$ if $\phi_i \in A$, otherwise $\delta_{\phi_i} = 0$, and $\phi_k$ is the random variable sampled from $\beta$ [124, 125]. For the dice example, $\phi_k$ is the face drawn after throwing the dice sampled from the bag of dice. In the literature, $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots)$ is said to follow a Griffiths-Engen-McCloskey distribution, i.e. $\mathrm{GEM}(\gamma)$ when $a_k$ is distributed according to a $\mathrm{Beta}(1, \gamma)$ with parameter $\gamma$. $\gamma$ is known as the concentration parameter [47].

Equation (3.15) assigns a probability to a set of the infinite sample space by the use of an indicator, i.e. the Dirac function, and a probability weight $\pi_k$, where $\sum_i^{\infty} \pi_k = 1$. In the analogy of dice with an infinite number

of faces, $\pi_k$ is the probability that a face $\phi_k$ is drawn. If we define $A$ as the set of even numbers, then $P(A)$ is the probability of drawing an even number, and $P(A^c)$ is the probability of odd numbers, the complement of $A$. The vector $(P(A), P(A^c))$ is Dirichlet distributed with parameters $(\alpha\boldsymbol{\beta}(A), \alpha\boldsymbol{\beta}(A^c))$, where $\alpha$ and $\beta$ are parameters of the Dirichlet Process. Equation (3.15) and the latter property of the vector $(P(A), P(A^c))$ are equivalents. A more detailed proof is given by Ferguson [35]. This property affirms that for any finite measurable partition of the infinite sample space, i.e. $\{A_i\}_{i=1}^k$, the vector $P(A_1, \cdots, A_k)$ has a Dirichlet distribution with parameters $(\alpha\boldsymbol{\beta}(A_1), \cdots, \alpha\boldsymbol{\beta}(A_k))$. One can make an analogy of this theorem with the aggregation property of the Dirichlet distribution, when aggregating the infinite "atoms" $\delta_{y_i}$ in finite partitions of the infinite space.

The parameters $\boldsymbol{\beta}$ and $\alpha$ play an important role in the behaviour of the process. The base distribution $\boldsymbol{\beta}$ has a similar role to the mean $\mu$ parameter of a Gaussian distribution. Numbers sampled from a Gaussian distribution are close to its mean with high probability, and far from the mean with low probability. In other words, the sampled values tend to fall around the mean. Similarly, a distribution sampled from a Dirichlet process is sampled around the base distribution $\beta$. Moreover, $\alpha$ tells us how confident we are on the base distribution. For larger values of $\alpha$, we have more confidence in the base distribution, and the sampled distribution is closer to $\boldsymbol{\beta}$, and therefore, the variance is smaller. For smaller values of $\alpha$, we have less confidence in the base distribution, and the samples tend to fall further away from the base distribution, increasing its variance [124, 125]. We can again make the analogy with the Gaussian parameters. In this case, $\boldsymbol{\alpha}$ has a similar property to the standard deviation parameter of a Gaussian. It determines the spread of the process, or how much variance one can expect from different realizations. As an illustrative example, let the base distribution of a Dirichlet process be a Gaussian. The Dirichlet process draws a discrete set of probabilities, yielding a probability distribution that is close to a Gaussian distribution, and this proximity is scaled by $\alpha$. Figure 3.3 shows an example of three realizations of a Dirichlet process with base distribution $H$ equal to a Gaussian distribution of mean $\mu = 0$ and $\sigma = 1$.

There are three different values for $\boldsymbol{\alpha}$: $\alpha_i = 0.1$ for all $i$, $\alpha_i = 1$ for all $i$, and $\alpha_i = 10$ for all $i$. For all three cases, the sampled distribution is discrete and close to the base distribution, a Gaussian distribution with mean equal to zero and standard deviation equal to one. As discussed before, $\alpha_i$ represents the confidence in the prior. In other words, it represents how close to the base distribution the sampled distribution is. As one can see in Figure 3.3, the cumulative curves from the last graph are more similar than for the first graph.




(a) $\alpha_i = 0.1$ for all $i$. (b) $\alpha_i = 1$ for all $i$. (c) $\alpha_i = 10$ for all $i$.

**Figure 3.3:** Three samples of a Dirichlet process, whose base distribution is a Gaussian distribution with mean equal to zero and standard deviation equal to 1. The values of the Gaussian random variables are on the x-axis, and the cumulative probability function on the y-axis. For each realization, a different value of $\boldsymbol{\alpha}$ is used: (a) $\alpha_i = 0.1$ for all $i$, (b) $\alpha_i = 1$ for all $i$, and (c) $\alpha_i = 10$ for all $i$. For all of them, the sampled distribution is discrete. As $\alpha_i$ grows, the cumulative distribution of the sampled distribution gets closer to the cumulative function of the base distribution.

## 3.2 Infinite Hidden Markov Model

To better understand the infinite hidden Markov model, let's first review the finite hidden Markov model. A single-particle track that can be described by a finite hidden Markov model with $K$ states is allowed to switch among these states over time. Moreover, we assume that this switch can only happen at the observation times, i.e. during the time the particle is imaged

(a frame). The probability vector $\boldsymbol{\pi_{\sigma_k}}$ describes the probability of leaving state $\sigma_k$ to any other state. Further define $\boldsymbol{\pi_{\sigma_k}} = (\pi_{\sigma_k,1}, \pi_{\sigma_k,2}, \ldots, \pi_{\sigma_k,K})$. Each state $\sigma_k$, for $k \in 1, 2, \ldots, K$ yields spatial steps, corresponding to observed steps of the track, drawn from a Gaussian distribution with mean 0 and precision $\nu_{\sigma_k}$, where $k = 1, \ldots, K$. This Gaussian distribution represents the emission model, $F$, and describes the measure that each Markov state imprints on the observed data.

Experimental measurements give us the trajectory of each particle and thus the sequence of displacements, $\delta x_n$, for each particle, where $n$ is the frame index, and $n = 1, \ldots, N$. Furthermore, these displacements are described by the observations distribution defined for each diffusive state $\sigma_k$. Thus, the sequence of hidden states, $s = \{s_1, \ldots, s_n, \ldots s_N\}$, and the observed data are distributed according to:

$$\begin{aligned} s_n \mid s_{n-1}, \boldsymbol{\pi}_{s_{n-1}} &\sim Cat\left(\boldsymbol{\pi}_{s_{n-1}}\right), \\ \delta x_n \mid s_n, (\phi_k)_{k=1}^{\infty} &\sim F_{\phi_{s_n}}. \end{aligned}$$

The $K$-state HMM can be thought of as a dynamical Gaussian mixture [125], where each element of the sequence of hidden states, $s_k$, is a Gaussian mixture itself, and the value of $s_k$ defines the row of the transition matrix $\boldsymbol{\pi_{s_k}} = (\pi_{s_k,1}, \ldots, \pi_{s_k,K})$. Each element of $\boldsymbol{\pi_{s_k}}$ gives the weight of each possible state for the next state $s_{k+1}$. Thus, each Gaussian mixture is linked through the set of states. This set represents the possible states and needs to be the same for all the displacements.

The infinite Hidden Markov model is similar to the $K$-state HMM when $K$ is taken to infinity. Thus, the model relies on an unbounded set of states. The Dirichlet process is a nonparametric framework that allows for an unbounded set of states. For each value of the sequence of hidden states, we have a Dirichlet process describing the transition probabilities. However, since the states throughout all displacements are coupled, these Dirichlet processes need to be connected. To introduce this link between the Dirichlet processes, a Hierarchical Dirichlet process is used, where the atoms

associated with the state-conditional Dirichlet process are shared [125].

The Hierarchical Dirichlet process defines a random probability distribution $\boldsymbol{\pi}_{\boldsymbol{s_k}}$ for each state and a global probability distribution $\beta$ that is shared among the states. This base distribution is a Dirichlet process itself with concentration parameter $\gamma$. Using the stick-breaking construction formalism as defined in Section 3.1, one can represent the hierarchical Dirichlet process as well. In this representation, the parameters are distributed as follows:

$$\begin{aligned} \tilde{\beta} &\sim \text{GEM}(\gamma), \\ \tilde{\pi}_{\sigma_k} &\sim DP(\alpha, \tilde{\beta}), \\ \phi_{\sigma_k} &\sim H, \end{aligned}$$

where GEM($\gamma$) is the Griffiths-Engen-McCloskey process, i.e. stick-breaking construction, with concentration parameter $\gamma$, and DP($\alpha, \tilde{\beta}$) is the Dirichlet process with concentration parameter $\alpha$ and base distribution $\tilde{\beta}$, and $H$ is the prior distribution for $\phi_{\sigma_k}$[5, 38, 124, 125, 131].

## 3.3 Beam Sampler

The infinite number of states in the Markov process yields infinitely many possible sequences of states. Thus, one cannot use the forward-backward algorithm to sample state trajectories [100, 115, 131], since there are infinitely many calculations to perform. The Beam sampler is an algorithm that makes sampling the state trajectory of an infinite hidden Makov process possible. This method was developed by Gael et al.[131] to be used on the infinite hidden Markov process. This section is an explanation of the sampler, based mostly on their work.

The main feature of the Beam sampler is that it introduces an auxiliary variable $\boldsymbol{u}$ such that the probability of a state trajectory conditioned on $\boldsymbol{u}$ is positive only for a finite number of trajectories. With a finite number of state trajectories, we can then sample a whole sequence of states using dynamic programming.

Let $\boldsymbol{s}$ be a state trajectory, $\boldsymbol{\pi_k}$ the transition probabilities leaving state $k$, and $t$ be the time index representing a time point on the data. An auxiliary variable $u_t$ is introduced for each $t$, where $u_t$ is distributed accordingly an uniform distribution over the range $[0, \pi_{s_{t-1},s_t}]$, i.e. $u_t \sim \text{Uniform}(0, \pi_{s_{t-1},s_t})$ depending on $s_t, s_{t-1}$, and $\pi_{s_{t-1},s_t}$. The trajectories $\boldsymbol{s} = (s_1, \ldots, s_n)$ whose conditional probability on $\boldsymbol{u}$ is non-zero are the ones where $u_t \leq \pi_{s_{t-1},s_t}$.

Thus, we sample the state trajectory $\boldsymbol{s}$ using a dynamic programming algorithm called the forward filtering-backward algorithm. The first step of the algorithm calculates $p(s_T|y_{1:T}, u_{1:T})$, the probability of having the state $s_T$ given the data points $y$ and the auxiliary variable for each time point on the data. Next, $s_T$ is sampled from $p(s_T|y_{1:T}, u_{1:T})$. Next, through a backward pass, $s_t$ is sampled conditioned on $s_{t+1}$ for all $0 \leq t \leq T-1$ in the data.

$p(s_T|y_{1:T}, u_{1:T})$ is calculated iteratively using a forward iteration. First, note that the probability of $s_t$ conditioned to the data and auxiliary variables up to $t$ is proportional to the joint probability of $s_t, y_t, u_t$ conditioned on the data and auxiliary variables up to $t-1$,

$$p(s_t|y_{1:t}, u_{1:t}) \propto p(s_t, u_t, y_t|y_{1:t-1}, u_{1:t-1}). \tag{3.16}$$

Therefore, we have that:

$$\begin{aligned} p(s_t,u_t,y_t|y_{1:t-1},u_{1:t-1}) = \sum_{s_{t-1}} & p(y_t|s_t)p(u_t|s_t, s_{t-1}, \boldsymbol{\pi_{s_{t-1}}})p(s_t|s_{t-1}) \\ & p(s_{t-1}|y_{1:t-1}, u_{1:t-1}) \end{aligned} \tag{3.17}$$

From the definition of the auxiliary variables, we obtain the conditional probability density of $u_t$ given the states and transition probability.

$$p(u_t|s_t, s_{t-1}, \boldsymbol{\pi}) = \frac{\mathcal{I}\left(0 < u_t < \pi_{s_{t-1},s_t}\right)}{\pi_{s_{t-1},s_t}}, \tag{3.18}$$

where $\mathcal{I}(C)$ is the indicator function and is equal to 1 if condition $C$ is true and equal to zero otherwise. Therefore equation (3.18) becomes

$$p(s_t,u_t,y_t|y_{1:t-1},u_{1:t-1})=p(y_t|s_t)\sum_{s_{t-1}} p(u_t|s_t,s_{t-1},\boldsymbol{\pi})\mathcal{I}\big(0<u_t<\pi_{s_{t-1},s_t}\big), \tag{3.19}$$

since $p_{s_{t-1},s_t} = \pi_{s_{t-1},s_t}$. The summation $\sum_{s_{t-1}}$ on equation (3.19) is over an infinite number of terms, since there an infinite number of possible states for $s_{t-1}$. However, this sum is truncated based on two conditions $p\left(u_t|s_t, s_{t-1}, \boldsymbol{\pi}\right) >$ zero, and $u_t < \pi_{s_{t-1},s_t}$, yielding a finite number of terms. This happens because the auxiliary variable divide the set of possible transition probabilities leaving state $s_{t-1}$ into two sets: the one with $\pi_{s_{t-1},k} < u_t$ and the one with $\pi_{s_{t-1},k} > u_t$. Since $\sum_k \pi_{s_{t-1},k} = 1$, the first set has an infinite number of elements, whereas the second one has a finite size.

Using (3.19) we calculate $p(s_T|y_{1:T}, u_{1:T})$, and then sample $s_T$. Finally, we sample the whole state trajectory $\boldsymbol{s}$ from $p\left(s_t|s_{t+1}, y_{1:t}, u_{1:t}\right)$ using the following iterative backward pass conditioned on $s_{t+1}$.

$$p\left(s_t|s_{t+1}, y_{1:t}, u_{1:t}\right) \propto p\left(s_{t+1}|y_{1:t}, u_{1:t}\right) p\left(s_{t+1}|s_t, u_{t+1}\right). \tag{3.20}$$

After sampling the whole sequence of states, the algorithm goes on to sample the base distribution, the transition probability matrix and the parameters of the emission model. The following sections explains these next steps.

Another possible approach for the estimation of number of states is using the weak limit of nonparametrics, where the number of states is fixed to be a large number. In this case, the Dirichlet process becomes a Dirichlet distribution with a large number of components in its base distribution, and this base distribution is still a random variable following a stick-breaking construction. This approach would be more computationally efficient than the nonparametrics Bayesian framework, where we allow for an infinite-dimensional state space instead fixing it to a large number.

## 3.4 Emission Model

The emission model is the model describing the relation of the data to the Markovian states. For this work we model single particle tracking displacement with a pure Brownian motion. Therefore, a Gaussian distribution is used to describe the data. The likelihood of the displacements $\boldsymbol{y}$ given the parameters of the model is a summation of Gaussian distributions with mean zero and precision $\nu$.

$$p\left(\boldsymbol{y}|\nu\right) = \left(\frac{\nu}{2\pi}\right)^{N/2} \exp\left(-\nu \sum_{i=1}^{N} \frac{y_i^2}{2}\right), \tag{3.21}$$

where $N$ is the number of displacements on the data, i.e. size of $\boldsymbol{y}$. The posterior of the emission model is equal to the likelihood given by equation (3.21) and the prior of the parameters. In order to have a closed form for the posterior, we can set the prior to be a conjugate of the likelihood. The conjugate prior for a Gaussian distribution with known mean and precision $\tau$ as the model parameter is a Gamma distribution.

From Bayes' theorem we know that the posterior is proportional to the likelihood times the prior. If the prior $p(\nu)$ of the precision parameters is a Gamma distribution with shape parameter $a$ and scale parameter $b$, the posterior is given by:

$$\begin{aligned} p\left(\nu|\boldsymbol{y}\right) \propto & p\left(\boldsymbol{y}|\nu\right) p\left(\nu\right) \\ &= \left(\frac{\nu}{2\pi}\right)^{N/2} \exp\left(-\nu \sum_{i=1}^{N} \frac{y_i^2}{2}\right) \frac{b^a \nu^{a-1} e^{-b\nu}}{\Gamma(a)} && (3.22) \\ &\propto \nu^{a+\frac{N}{2}-1} \exp\left(-\nu \left(b + \frac{\sum_{i=1}^{N} y_i^2}{2}\right)\right) && (3.23) \end{aligned}$$

Therefore, in each iteration the algorithm samples the parameter $\nu$ from a Gamma distribution with shape parameter equal to $a + \frac{N}{2} - 1$ and scale

parameter equal to $b + \frac{\sum_{i=1}^{N} y_i^2}{2}$.

## 3.5 iHMMSPT algorithm (SPT-$\infty$)

The algorithm samples the posterior distribution of the parameter set given the observations. For the infinite hidden Markov model for single-particle trajectories (iHMMSPT) algorithm, the observations are single-particle tracks displacements, and the parameters are the number of states, the transition probabilities, and the emission parameters for each state. In this thesis, we refer to iHMMSPT as SPT-$\infty$.

The following steps summarize the algorithm:

A Generate auxiliary variable for each time point of the dataset, i.e. $\boldsymbol{u}^{(l+1)} = (u_1, \ldots, u_N)$ using $u_t \sim \mathcal{U}\left(0, \pi_{s_{t-1} \to s_t}\right)$.

B Use $\boldsymbol{u}^{(l+1)}$ to expand the state space $\mathbb{S}$.

C Generate the new sequence of states $\boldsymbol{s}^{(l+1)}$ using the forward-filtering backward algorithm.

D Compress the state space $\mathbb{S}$, excluding the states that are not visited in $\boldsymbol{s}^{(l+1)}$.

E Generate $\tilde{\nu}^{(l+1)}$ using the posterior distribution.

F If in burn-in phase ($(l+1) \leq$ burn-in), compress $\mathbb{S}$ by merging states using Bd (explained below). Update $\boldsymbol{s}^{(l+1)}$.

G Generate $\tilde{\beta}^{(l+1)}$ using the new state sequence.

H Generate the transition probability matrix $\tilde{\tilde{\pi}}^{(l+1)}$ using $\tilde{\beta}^{(l+1)}$ and $\boldsymbol{s}^{(l+1)}$.

Step B of the above algorithm is related to the beam sampler. The auxiliary variable $\boldsymbol{u}$ and the condition that $u_t \leq \pi_{s_{t-1}, s_t}$ for all $t$ truncates the number of possible hidden sequences of states to a finite number. A

consequence of that is the truncation of the state space as well. As the state space grows in size, the transition probability vector became more sparse, increasing the number of states where the transition probability to it is larger than $u_t$ resulting in a zero probability for a trajectory with that state on $t$ existing. Therefore, it is possible to define a number of meaningful states that add information to the algorithm [131, 137].

For a more detailed explanation, let $\mathbb{S}$ be the state space, where its size is equal to $K$. The auxiliary variable allows for an expansion in $\mathbb{S}$ only when needed. Define the auxiliary variable to be $u_t$, for each time point $t = 1 \cdots N$ of a track. The $u_t$ are drawn from a uniform distribution defined as: $u_t \sim \mathcal{U}\left(0, \pi_{s_{t-1} \to s_t}\right)$. Then, the probability of a transition from state $\sigma_k$ in $\mathbb{S}$ to any state outside $\mathbb{S}$ is equal to

$$P_{exit\,\sigma_k} = \left(1 - \Sigma_{m=1}^{K} \pi_{\sigma_k \to \sigma_m}\right).$$

If the condition $\max\limits_{1 \le k \le K} P_{exit\,\sigma_k} > \max\limits_{1 \le i \le N} u_i$ holds, then we do not have all the necessary states in the state space $\mathbb{S}$ to explain the data, since the maximum probability of leaving the state space is larger than the maximum probability of transitioning between states inside the set $\mathbb{S}$. Thus, in this case we must add a new state in $\mathbb{S}$, whose size will increase to $K + 1$ states. States will be added to $\mathbb{S}$ until the condition fails.

After the state space has been defined, the sequence of states $s = \{s_i | i = 1, \ldots, N\}$ is sampled by using forward-filtering backward sampling on Step C.

Next, with the sequence of states for the trajectory, we can check whether there are any state of the state space $\mathbb{S}$ that has not been visited. Any state that has not been visited is deleted from the state space $\mathbb{S}$. This step is called compression of space state. In summary, we expand the state space to avoid underfitting, and we compress $\mathbb{S}$ to avoid overfitting. Finally, we sample the parameters $\tilde{\beta}, \tilde{\pi}_{s_t}$, and the emission parameters $\nu_{\sigma_k}$ from the associated posterior distribution as in equation (3.23).

In summary, given a sequence of states $\boldsymbol{s}^{(l)}$, a sample of $\tilde{\beta}^{(l)}$, a transition probabilities matrix $\tilde{\tilde{pi}}^{(l)}$, and emission parameters $\tilde{\nu}^{(l)}$, where $l$ is the itera-

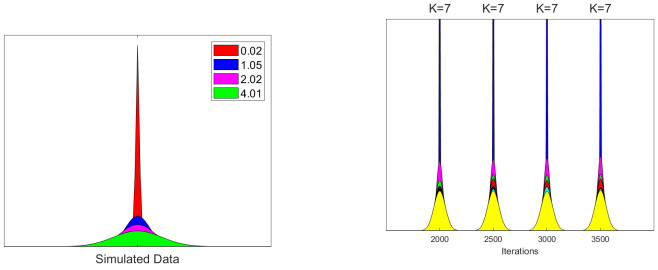
tion index, and using a Markov-Chain Monte Carlo algorithm, the SPT-$\infty$ generates a new set of samples from $P\left(\boldsymbol{u}, \boldsymbol{s}, \tilde{\beta}, \tilde{\tilde{\pi}}, \tilde{\nu} | \boldsymbol{\Delta x}\right)$ following the steps from above.

Finally, after obtaining the number of states of the Markov model and a distribution of parameters of the Markov model, one might be interested in estimating the state that the particle occupies at each time point. To do this, we use the forward-backward algorithm to calculate the likelihood of being in state $\sigma_k$ given the number of states, their parameters, and the observations. We then select the state that gives the maximum likelihood in that specific time point [100].

### 3.5.1 Improving convergence by accelerating state space compression

In the iHMM algorithm described above, the compression of state space is a step that discards unnecessary states. After sampling the sequence of states, any state that has not been visited is discarded. Since they have not been visited, they did not produce any of the observations. However, we propose that it is helpful to be more strict with the SPT data. For our model, each diffusive state follows a Gaussian process with mean zero. This generates substantial overlap among the distributions of each state. Thus, the creation of a new state with diffusion coefficient close to a previously existing diffusion coefficient becomes common and this slows convergence. This is illustrated in Figure 3.4. In this figure, we first plot the Gaussian distribution for each state for a model with $K_{\text{real}} = 4$ states. We then iterate the SPT-$\infty$ algorithm using the usual compression method and plot the estimated Gaussian distribution for each state after 2000, 2500, 3000 and 3500 iterations (Fig. 3.4b). We can see that even after 3500 iterations, we do not achieve the correct number of states. However, if we increase the number of iterations from 3500 to 100000, the algorithm does converge (Fig. 3.5).

Seeking to accelerate convergence, we decided to add another condition, only at the burn-in phase, to additionally compress the state space. As




(a) Simulated data with $K = 4$ states.

(b) Estimated $K = 7$ states for simulated data with only $K_{\text{real}} = 4$ states.

**Figure 3.4:** Graphs of the Gaussian distribution for each state: (a) for simulated data with $K_{\text{real}} = 4$ states with diffusion coefficients as indicated, and (b) of estimated states using iHMMPST over 3500 iterations. The estimated number of states is $K = 7$, and we can see a lot of overlap among the Gaussians distributions of each state.

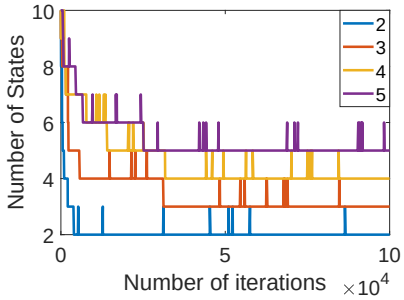



**Figure 3.5:** Convergence of algorithm without any additional conditions in the compression of states step. We ran the algorithm for 4 different sets of simulated data (2, 3, 4 and 5 states). We can see that the algorithm converges to the correct number of states for all cases, after around $4 \times 10^4$ iterations.

a remark, convergence here is assessed informally, and it is not referring to the usual formal notion of MCMC convergence such as the Geweke's test [46, 128] for single MCMC chains and the Gelman-Rubin convergence diagnostic for multiple MCMC chains [42, 128]. We use the Bhattacharya distance (Bd), a quantity that measures the similarity between two probabil-

ity distributions, to decide if two states should be merged into one. For two Gaussian probability distributions with means equal to zero and precisions $\nu_1$, $\nu_2$, the Bhattacharya distance is equal to

$$Bd = \sqrt{\frac{2\sqrt{\nu_1\nu_2}}{\nu_1 + \nu_2}}. \tag{3.24}$$

For equal precision, $\nu_1 = \nu_2$, we have $Bd = 1$, and as $\nu_1$ and $\nu_2$ grow apart, $Bd \to 0$.

We plot $Bd$ as a function of the precision of two Gaussian distributions in Figure 3.6. We observe that $Bd$ is higher than $\sim 0.5$ for the majority of the domain.

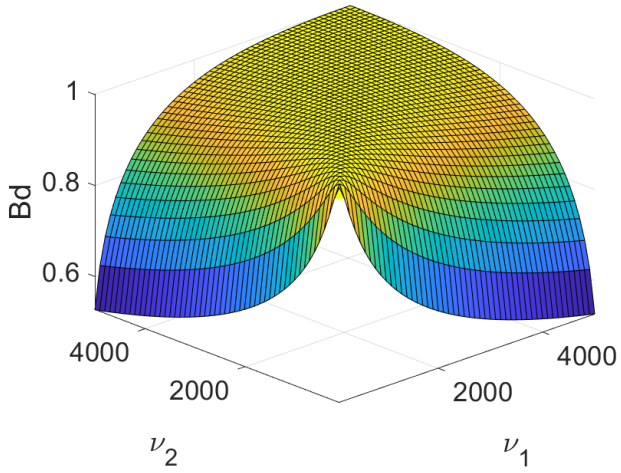



**Figure 3.6:** Bhattacharya distance for two Gaussian distributions with same mean, and precisions $\nu_1$ and $\nu_2$.

In the iHMM algorithm, we merge two different states into one if their pairwise $Bd$ exceeds a given threshold. For simulated data, the threshold we use is $1-10^{-6}$. Since simulated data is a perfect representation of the model, we choose a very large threshold to make sure no valuable information is lost.

Another point to take into consideration is that Dirichlet processes are not consistent, which leads to a different number of states for datasets with

same number of states [81]. Therefore, an additional step that merges similar states could tackle this issue.

We perform a comparison between the algorithm with and without the Bhattacharya distance condition. In Figure 3.7, we plot the number of states against the iteration number. The algorithm without Bhattacharya takes more than $10^4$ iterations to achieve the correct (simulated) number of states, whereas the algorithm with the additional condition takes approximately $10^2$ iterations to arrive at the correct number of states. Thus, we have at least a 2 order of magnitude decrease in time to convergence if the additional condition is used. Moreover, the distribution of diffusion coefficient estimates of each state for each algorithm is very similar, as one can see in Figure 3.8.

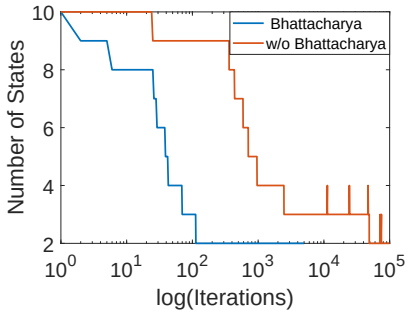



**Figure 3.7:** Semilog plot of the number of states for each iteration of two algorithms: SPT-$\infty$ without Bhattacharya distance as a condition to merge states (red curve), and SPT-$\infty$ with the Bhattacharya condition (blue curve), both applied to synthetic data with two states. The additional condition reduces the required number of iterations for convergence by a factor of around 500.

To determine the threshold for experimental data, we use a heuristic based on the localization accuracy of the data, as follows. Let $\varepsilon$ be the measured localization accuracy of the experiment. We want to estimate a lower bound for the $Bd$ threshold so that the differences between estimated diffusions exceed the accuracy of our experiment. Let $\nu_1$, and $\nu_2$ be the precisions of two states. If the difference between the variance of these

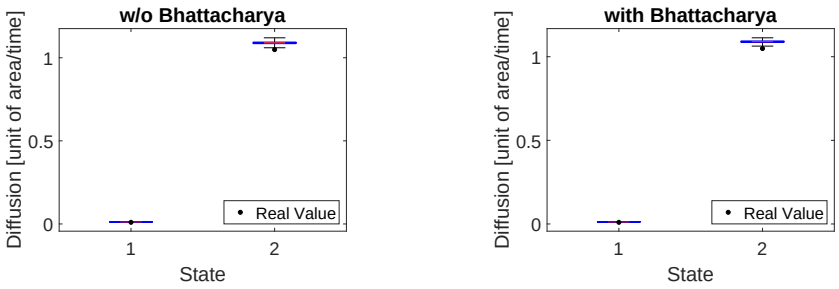



**Figure 3.8:** Diffusion coefficient distribution estimates for each algorithm: (a) SPT-$\infty$ without the Bhattacharya condition; (b) SPT-$\infty$ with the Bhattacharya condition. Black dots indicate the true (simulated) value of the diffusion coefficient for each state. In both cases, the estimation is in good agreement with the true value.

states is within the experimental accuracy, then variance of the second state has to be $\frac{1}{\nu_2} = 2D_2\tau \sim 2D_1\tau + 2\varepsilon^2$, with $\nu_1 = \frac{1}{2D_1\tau}$. The pairwise $Bd$ for states with difference in the diffusion coefficients within the accuracy of the experiment, $Bd_{acc}$, is given by:

$$Bd_{acc_{\nu_1}} = \sqrt{2}\sqrt{\frac{\sqrt{\nu_1\left(\frac{1}{\nu_1}+2\varepsilon^2\right)^{-1}}}{\nu_1+\left(\frac{1}{\nu_1}+2\varepsilon^2\right)^{-1}}}. \tag{3.25}$$

Using equation 3.25, we define a dynamical threshold for the experimental data. For each pair of states, $(m, n)$, we calculate $Bd_{acc_{\nu_m}}$ and $Bd_{acc_{\nu_n}}$. Thus, the threshold for this pair of states, $(m, n)$, is given by $\max(Bd_{acc_{\nu_m}}, Bd_{acc_{\nu_n}})$.

The guarantees of convergence for the MCMC algorithm should be nullified with the addition of the merging states step, and as a consequence coverage properties of the credible interval for each parameter should also be affected. In addition, the detailed balance is not satisfied any more, given that states are only merged and are not separated.

## 3.6 Results

### 3.6.1 Algorithm testing with simulated data

We first simulate trajectories of particles whose motion is defined by steps drawn from a set of possible diffusive states. Particles transition between states with fixed rates, forming a Markov process. We then analyze the simulated trajectories with the SPT-$\infty$ algorithm, to obtain estimates of the number of states, the diffusion coefficient of each state and the transition matrix of the Markov Model. Table 3.1 summarizes results for five example datasets, each with a different number of states. Each simulated particle is simulated in two dimensions, over $5 \times 10^4$ frames, with a simulated frame-rate of 1000 frames per second.

The algorithm is divided into two phases: a burn-in phase where we iterate the algorithm 750 (15%) times using the Bhattacharya condition to compress the state space, followed by 4250 (85%) further iterations using the usual compression step (Methods). After burn-in, the iterations of the process provide estimated distributions for number of states, diffusion coefficients and state transition matrix. In Table 3.1, we show the mode of the estimated number of states, the mean of the distribution for each diffusion coefficient and the mean of the stationary probability for each inferred state.

In Figure 3.9, we show the convergence of the number of states for each example. For each dataset, we ran the SPT-$\infty$ 10 times. The initial number of states was set to 10 for all runs. This allows us to assess the ability of the algorithm to reliably converge to the correct parameters, including the number of states. We observed that the algorithm converges rapidly, and in the majority of cases converges to the correct number of states. For datasets with more than 2 states, the algorithm did not converge to the correct number of states for all 10 runs. This is reasonable, given that more states yields more parameters in the model, adding complexity to the model, and slowing the convergence. This indicates that there is a trade-off between fast convergence and guaranteed accuracy.

The diffusion coefficients are also key parameters of the model. In Figure 3.10, we show how the sampled diffusion coefficients changed through the

iterations. We observe the initial exploration of the algorithm on different choices of parameters until it converges to a ground truth of five different values. Moreover, in Figure 3.11 we show the final distributions obtained for the diffusion coefficients for each simulated dataset for a single run. We observed that the estimated diffusion coefficients are in good agreement with those used for simulation. Moreover, as we expect, the variance of the final distribution is inversely correlated with the stationary probability of each state. State four of the five-state simulated dataset is a good example of this (see also Table 3.1).

| Simulated | | | Estimated | | |
|---|---|---|---|---|---|
| # of states | D | SP | K | D | SP |
| **5** | 0.0466 | 0.2422 | **5** | 0.0549 | 0.2523 |
| | 1.0353 | 0.1784 | | 1.1359 | 0.1546 |
| | 2.2200 | 0.2187 | | 2.3033 | 0.2748 |
| | 4.8702 | 0.1383 | | 6.5256 | 0.1365 |
| | 10.680 | 0.2225 | | 11.162 | 0.1818 |
| **4** | 0.0113 | 0.2118 | **4** | 0.0177 | 0.2299 |
| | 1.0456 | 0.2286 | | 1.3332 | 0.1655 |
| | 2.2146 | 0.3247 | | 1.9003 | 0.3617 |
| | 4.8789 | 0.2349 | | 4.8467 | 0.2428 |
| **3** | 0.0457 | 0.5174 | **3** | 0.0494 | 0.5302 |
| | 1.0503 | 0.1341 | | 1.3529 | 0.1650 |
| | 2.2113 | 0.3485 | | 2.2475 | 0.3048 |
| **2** | 0.0098 | 0.4552 | **2** | 0.0124 | 0.4736 |
| | 1.0489 | 0.5448 | | 1.0896 | 0.5264 |

**Table 3.1:** Results from testing the SPT-$\infty$ algorithm with simulated data. We simulated four different datasets with 2-5 diffusive states. Here, we report the mode of the estimated number of states after burn-in (K), the mean estimated diffusion coefficient for each distribution (D), and the mean stationary probability (SP). The stationary probability is calculated from the estimated transition rate matrix. All quantities have appropriate arbitrary units.

We observe from Table 3.1 that the algorithm estimates the correct (simulated) number of states $K$ for every dataset. We also obtain generally good

estimates of the diffusion coefficients. The signs of errors in estimation of diffusion coefficients are in line with our expectations. For example, for $K = 4$, we have two states with true diffusion coefficients equal to 1.05 and 2.21 in appropriate units. The estimated diffusion coefficients are 1.33 and 1.90 in appropriate units, respectively. These discrepancies occurred because some displacements that are actually from the slower states have been treated as displacements from the faster state. We can confirm this by looking at the stationary probabilities. The faster state stationary probability was overestimated, while the stationary probability of the slow state was underestimated. However, in general, we find that the algorithm performs well with simulated data of this type, even though the number of parameters to be estimated is quite large. For example, when $K = 4$, we have 20 parameters to estimate: four diffusion coefficients and 16 transition probabilities.

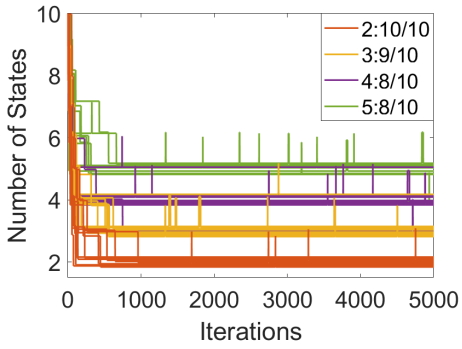



**Figure 3.9:** Number of states estimates at each iteration for each simulated dataset using the additional compression criterion. Ten different runs of the algorithm are shown for each set of simulated data (with colours indicated). Convergence was generally achieved after about 1000 iterations. 8 out of 10 runs converged to the correct number of states for the four- and five-states dataset, 9 out of 10 runs converged correctly for the three-states dataset, and all 10 runs converged correctly for the two-state dataset.

Finally, we performed state segmentation on each simulated trajectory

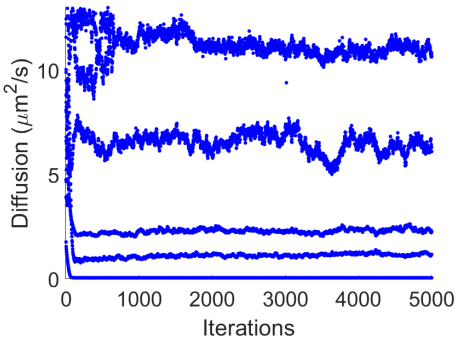
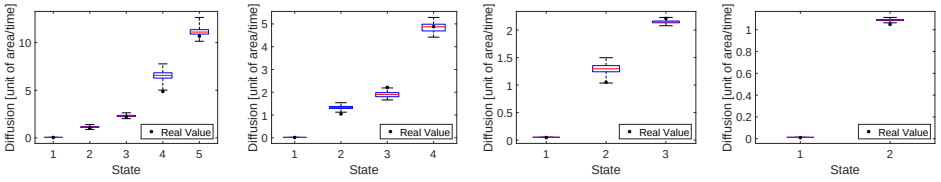



**Figure 3.10:** Diffusion coefficient estimates at each iteration of the algorithm, using simulated four-state data. There is a transition from five to four estimated states just before iteration 1000.




**(a)** 5 states **(b)** 4 states **(c)** 3 states **(d)** 2 states

**Figure 3.11:** Estimated diffusion coefficient distributions for simulated data with different numbers of states. Black circles represent true diffusion coefficients.

by finding the most probable state at each frame, given the estimated parameters for that trajectory. We then compared the estimated state sequence with the true simulated state sequence and calculated the percentage of displacements that are correctly classified (Table 3.2). We observed that the accuracy of the state segmentation decreases as the number of states grows. This is expected because the quality of the diffusion and transition parameter estimates generally decreases as the number of states increases, and this affects the quality of state segmentation.

We also analyzed the performance of the algorithm with the presence of noise in the data. We simulated particles as before: steps were drawn from possible diffusive states, but we then added Gaussian noise to represent local-

| Number of States | Correct State Segmentation |
|---|---|
| 5 | 61% |
| 4 | 62% |
| 3 | 84% |
| 2 | 97% |

**Table 3.2:** Accuracy of state segmentation from simulated trajectories.

ization uncertainty. Here, we simulated 100 particles in 2 dimensions, each with 30000 frames, and a simulated frame-rate of 33 frames per second. For simplicity, we performed simulations of three-state models with 5 different levels of noise: $0, 0.001, 0.005, 0.01$ and $0.05$ unit of length. Summary results are shown in Table 3.3. The algorithm was able to find the correct number of states for the first 3 levels of noise, and to recover the diffusion coefficient and stationary probability of each state. However, when the noise level gets large enough, the algorithm fails to find the correct number of states and/or the diffusion coefficient and stationary probability of each state. Approaches to correct for localization errors in mobility modelling have been described by others [6, 58, 68, 101, 121, 135]. In Chapter 4, we further generalize the SPT-2E model by developing a novel framework to estimate the number of states and their parameters simultaneously.

In this work, we only tested the algorithm on simulated pre-localized traces from a multi-state measurement process. These trajectories were used to analyse the performance of the SPT-$\infty$ algorithm. There are potential limitations related to this approach. In practice, the experimental trajectories are subject to different noise and errors due to the pixelated point-spread function, detector noise, etc, and the measurement process is a model that potentially captures these errors. Therefore, the measurement process and the algorithm are not fully tested by simulated traces. Moreover, the experimental trajectories are also subject to detection and tracking algorithm errors, and by simulating pre-localized traces, the algorithm has not been tested against these additional imprecisions. To further test the algorithm, one could simulate the raw data originated from a microscopy, allowing for

| **Simulated** | **K** | $D_1$ | $SP_1$ | $D_2$ | $SP_2$ | $D_3$ | $SP_3$ |
|---|---|---|---|---|---|---|---|
| | **3** | **0.05** | **0.2788** | **0.2** | **0.2360** | **0.5** | **0.4853** |
| ***s.t.d.* noise** | *Estimated* | | | | | | |
| **0** | 3 | 0.0484 | 0.2626 | 0.1789 | 0.2366 | 0.4919 | 0.5008 |
| **0.001** | 3 | 0.051 | 0.2893 | 0.2068 | 0.2385 | 0.5020 | 0.4721 |
| **0.005** | 3 | 0.0493 | 0.2870 | 0.2064 | 0.2256 | 0.5041 | 0.4873 |
| **0.01** | 3 | 0.0559 | 0.3008 | 0.204 | 0.2071 | 0.5034 | 0.4922 |
| **0.05** | 2 | 0.1451 | 0.396878 | 0.5431 | 0.6032 | − − − | − − − |

**Table 3.3:** Results from testing the SPT-$\infty$ algorithm with noisy simulated data. We simulated five different datasets with three diffusive states, and five different level of added noise. The standard deviation of each added noise was:0, 0.001, 0.005,0.01, and 0.05 Here, we report the mode of the estimated number of states after burn-in (K), the mean estimated diffusion coefficient for each distribution (D), and the mean stationary probability (SP). The stationary probability is calculated from the estimated transition rate matrix. We ran our algorithm five times on each dataset. The results among the different runs were all similar, obtained the same optimal number of states and similar values of D and SP.

the incorporation of more realistic noise such as pixelated point-spread function, detector noise, etc. Through this approach, both the measurement process and SPT-$\infty$ could be better evaluated.

### 3.6.2 Application to B cell receptor tracking data

We now apply the SPT-$\infty$ algorithm to experimental data. In previous work, we performed a detailed comparison of results from SPT using two different methods for labeling cell-surface proteins. Briefly, B cell receptors (BCR) on the surface of B lymphocytes were labeled using quantum dots (Qdots) linked to monovalent antigen-binding fragments of antibodies (Fab), or with Fab fragments that were directly conjugated to the small organic fluorophore Cy3. Images were taken at 33 frames per second using a total internal reflection fluorescence (TIRF) instrument. Tracks were extracted from image stacks using Icy software [18, 27]. The spatial precision of par-

ticle localization was estimated to be 23nm and 30nm for Qdot and Cy3 labelling, respectively. Full methods are reported in our original paper [1].

Using these data, we previously conclusively showed that the mobility of proteins labeled using Qdots was impaired, most probably due to steric hindrance. As part of our analysis, we used a two-state HMM to segment the particle tracks among two diffusive states: a fast and a slow state [24]. Using the SPT-$\infty$ algorithm, we can now answer the question as to whether the data is better described by a different number of states. This approach has the potential to give us insights into the heterogeneity of potential interactions between the tracked receptor and different systems at the cellular membrane, such as the cortical actin cytoskeleton, lipid rafts, transiently binding proteins, etc. We can also compare the two labelling strategies and obtain a refined picture of changes in the diffusive behaviour.

We examined six datasets, each from a separate experiment. Three experimental datasets used directly-conjugated Cy3-labelling of IgG and three used Qdot-labelling of IgG [1]. For the Cy3 experiments, the datasets contained 1054, 1015 and 1040 tracks, respectively, with median number of frames per track of 51, 53 and 47 (standard deviation of 79, 80, 75). For the Qdot experiments, the datasets contained 1117, 1087 and 502 tracks, respectively, with median numbers of frames per track of 55, 62, 78(standard deviation 92, 95 and 102). Before applying our SPT-$\infty$ algorithm, we first applied an immobilility threshold to remove immobile tracks, as previously described [1]. We allow the algorithm to complete 10000 iterations in total, with a burn-in phase of 1500 iterations. For each trial, we ran the algorithm with the additional Bhattacharya condition chosen based on localization accuracy (Methods).

Results are shown in Tables 3.4 and 3.5, and in Figure 3.12. The diffusion coefficients and stationary probabilities reported are the mean of the distributions of each state of all iterations. We also report the estimated full transition matrices for each labelling method in the Appendix.

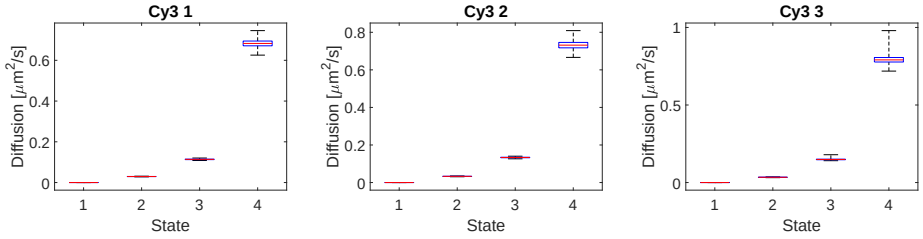
Interestingly, we find that the optimal number of states was four for all Cy3 experiments, and five for all Qdot experiments, except for the third trial of the Qdot-labelling experiment, where the optimal number of states

was four. We also found that the parameter estimates for each state were qualitatively similar from experiments ran on different days. Examining the states in detail, we see that the first state is very slow for all experiments, reflecting particles that are transiently in an immobile state (recall that entirely immobile particles are removed from the data before analysis). Meanwhile, the largest diffusion coefficients (fourth state) are very large, estimated to be on the order of $0.1 - 1\,\mu\text{m}^2\text{s}^{-1}$. The stationary probabilities for these show that they account for 5-6% of all states for Cy3-labelled receptors, but only 2-3% for Qdot-labelled receptors. Since Qdots are brighter and less prone to blinking than Cy3 molecules, this suggests that some or all of these larger steps are in fact due to tracking errors. We also found that the extra state for the Qdot molecules have diffusion coefficient of approximately $0.006\,\mu\text{m}^2\text{s}^{-1}$, which was much lower than that for the Cy3-labelling. The stationary probability for this state shows that it account for 20% of the Qdot-labelled receptors, indicating that Qdot labelling generally impairs protein mobility. In contrast, the most-occupied state for the Cy3-labelled molecules is the third state ($D \sim 10^{-1}\mu\text{m}^2\text{s}^{-1}$), whereas for the Qdot-labelled molecules, the most-occupied state is the slower third state ($D \sim 10^{-2}\mu\text{m}^2\text{s}^{-1}$). We also calculated the effective diffusion coefficient defined as $D_{\text{eff}} = P_1D_1 + P_2D_2 + \cdots + P_nD_n$. We obtain similar results to our previous work [1] for all datasets (not shown). Overall, our results are consistent with the hypothesis that the Qdot-labelled receptors can easily become trapped in small regions of the cell membrane, while the Cy3-labelled molecules can escape these regions and explore the cell membrane.

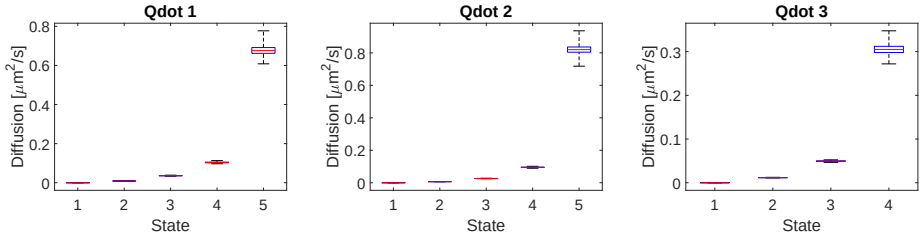
In Figure 3.13, we plot trajectories from two experiments, segmented (colour-coded) by diffusive state across the trajectory. Most Qdot trajectories reflect limited receptor mobility and are highly localized in very small regions, and mostly in the third state (red). On the other hand, the Cy3-labelled receptors are mostly in their (faster) fourth state (cyan) and are able to explore more of the cell surface. We can also observe transitions between states over time. We found that most particles underwent at least one transition during their trajectory. The proportion of particles that did not transition among states at least once was small ($1.3\% - 4.6\%$). However,

Qdot-labelled particles are slightly less likely to never undergo a transition when compared to Cy3-labelled molecules (Figure 3.14).

These results about the impaired mobility of Qdot-labelled receptors are in line with previous work [1]. However, we previously imposed a two-state model, while here we find that five states are optimal to describe the data. Even after possibly discarding the fastest inferred state as likely due to tracking errors, we find that receptor mobility is more heterogeneous than previously thought. In general, using a model with only two states is likely to cause the loss of potentially important information.

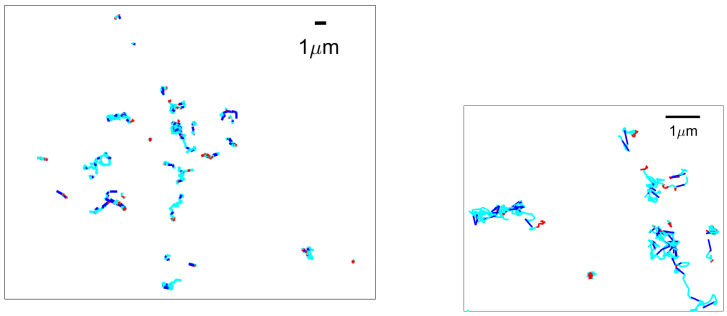



**(a)** Cy3

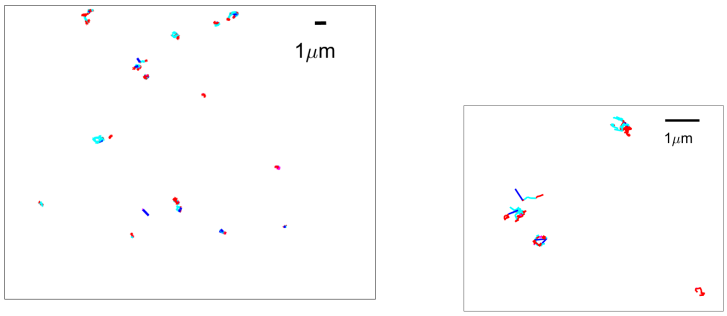



**(b)** Qdot

**Figure 3.12:** SPT-$\infty$ results from six sets of experimental trajectories obtained by labelling IgG receptors on the surface of live B cells (A20 cell lines) either using a Cy3-labelled probe or a Qdot probe.(a) Estimated diffusion coefficients of Cy3-labelled receptors. (b) Estimated diffusion coefficients of Qdot-labelled receptors.




**(a)** Cy3




**(b)** Qdot

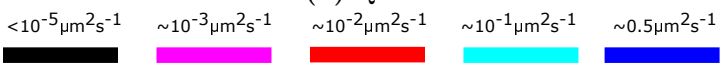



**Figure 3.13:** Segmented trajectories of (a) Cy3-labelled B cell receptors, and (b) Qdot-labelled B cell receptors, with a zoomed image of a smaller region. Colours represent the different diffusive states obtained from each experiment. Diffusion constant estimates are presented in Tables 3.4 and 3.5. The trajectories of Qdot-labelled molecules are spatially restricted compared to those of Cy3-labelled molecules. Moreover, Cy3 trajectories are estimated to frequently occupy the more-mobile states ($D \sim 10^{-1}\mu\text{m}^2\text{s}^{-1}$), whereas the Qdot trajectories are mostly found in slower states ($D \lesssim 10^{-2}\mu\text{m}^2\text{s}^{-1}$).

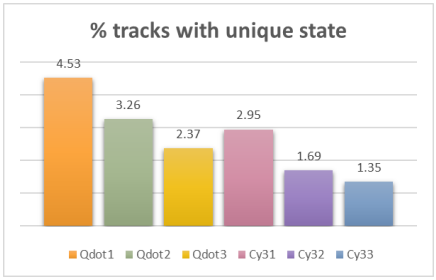



**Figure 3.14:** Proportion of tracks from each experimental dataset that never underwent a transition.

| ***Label*** | Cy3 1 | | Cy3 2 | | Cy3 3 | |
|---|---|---|---|---|---|---|
| $\Theta$ | $\mathrm{D}(\mu\mathrm{m}^2/\mathrm{s})$ | SP | $\mathrm{D}(\mu\mathrm{m}^2/\mathrm{s})$ | SP | $\mathrm{D}(\mu\mathrm{m}^2/\mathrm{s})$ | SP |
| **1** | $< 10^{-5}$ | 0.1253 | $< 10^{-5}$ | 0.1267 | $< 10^{-5}$ | 0.1272 |
| **2** | 0.0295 | 0.3044 | 0.0329 | 0.3365 | 0.0339 | 0.3358 |
| **3** | 0.1138 | 0.4980 | 0.1333 | 0.4821 | 0.1493 | 0.4713 |
| **4** | 0.6831 | 0.0723 | 0.7319 | 0.0547 | 0.7935 | 0.0657 |

**Table 3.4:** SPT-$\infty$E results from three sets of experimental trajectories, obtained by labelling IgG receptors on the surface of live B cells (A20 cell lines) using a Cy3-labelled probe. For each experiment, we estimate that four states is optimal. Mean estimated diffusion coefficients (D) and stationary probabilities (SP) are also reported for each state.

### 3.6.3 Effects of actin cytoskeleton disruption on BCR mobility

As a second case study, we re-examined a second set of data where the actin cytoskeleton of the B cell was disrupted by latrunculin A (LatA). Two datasets are generated by either Cy3-labelling of IgM BCRs on cells or Qdot-labelling on IgM BCRs that were treated with DMSO. These are the control experiments. The other datasets are generated by treating the sample with LatA [1]. For the Cy3 experiments, the datasets contained 1960 and 1661 tracks, respectively, with median number of frames per track of 45 (standard deviation 67). For the Qdot experiments, the datasets contained 3996 and 3889 tracks, respectively, with median number of frames per track of 54

| ***Label*** | Qdot 1 | | Qdot 2 | | Qdot 3 | |
|---|---|---|---|---|---|---|
| $\Theta$ | $D(\mu m^2/s)$ | Prob | $D(\mu m^2/s)$ | Prob | $D(\mu m^2/s)$ | Prob |
| **1** | $< 10^{-5}$ | 0.1281 | $< 10^{-5}$ | 0.1402 | $< 10^{-5}$ | 0.1255 |
| **2** | 0.0087 | 0.2166 | 0.0066 | 0.2267 | 0.0116 | 0.3635 |
| **3** | 0.0360 | 0.3938 | 0.0260 | 0.3871 | 0.0495 | 0.4551 |
| **4** | 0.1038 | 0.2334 | 0.0951 | 0.2187 | 0.3054 | 0.0559 |
| **5** | 0.6769 | 0.0280 | 0.8202 | 0.0273 | | |

**Table 3.5:** SPT-$\infty$E results from three sets of experimental trajectories, obtained by labelling IgG receptors on the surface of live B cells (A20 cell lines) using a Qdot probe. Experiments 1 and 2 were found to support five states, while experiment 3 supported four states. Mean estimated diffusion coefficients (D) and stationary probabilities (SP) are also reported for each state.

(standard deviation 82).

Results are shown in Table 3.6. For all datasets, we obtained 5 states as the optimal number of states, except for LatA-treated cells in which IgM was labelled with Cy3 Fab-anti-IgM, which had 4 states as the optimal number. We found very slow ($D \sim 10^{-7}\mu m^2 s^{-1}$) and very fast ($D \geq 1.1\mu m^2 s^{-1}$) diffusive states. Again, the slow state shows transient confinement, while the fast state probably reflects a certain fraction of tracking errors. Also as before, the fastest state was more heavily weighted in the Cy3 experiments, probably reflecting the fact they are less sterically hindered compared to Qdot experiments. Our results also show that labelled IgM is generally more mobile than labelled IgG, in agreement with previous analysis [1, 25].

For the Cy3 label experiments, we find interesting differences between control and LatA. In agreement with previous analysis, receptors on LatA-treated cells are generally much more mobile than on control cells, implicating the actin cytoskeleton as an important regulator of receptor mobility. However, we did not find any major differences between control and LatA conditions when using Qdot labels. This tells us that the steric hindrance of Qdots is sufficient to obscure the difference between control and LatA experiments on receptor mobility, given that the large Qdots decrease the diffusivity of surface receptors, which prevents an increase in mobility when

the actin cytoskeleton is distrupted. Comparing results between control and LatA treatment, LatA cells have only 4 states as opposed to 5 for control cells. The second slowest state is not present after LatA treatment, perhaps indicating the loss of small-scale constraints on IgM motion. Moreover, we find that the most-occupied state for the control cells is the third state ($D \sim 5 \times 10^{-2} \mu\text{m}^2\text{s}^{-1}$ in both cases), whereas for LatA, it is the fourth state ($D \sim 1.8 \times 10^{-1} \mu\text{m}^2\text{s}^{-1}$ in both cases), indicating an increase in receptor mobility via transitionS to more mobile states. Compared to two-state analysis, we obtained a refined picture of the heterogeneity of the system.

| ***Exp*** | Cy3 DMSO | | Cy3 LatA | |
|---|---|---|---|---|
| *States* | D($\mu$m$^2$/s) | Prob | D($\mu$m$^2$/s) | Prob |
| **1** | $< 10^{-5}$ | 0.1118 | $< 10^{-5}$ | 0.1130 |
| **2** | 0.0152 | 0.1985 | | |
| **3** | 0.0617 | 0.3675 | 0.0357 | 0.2743 |
| **4** | 0.1706 | 0.2635 | 0.1704 | 0.4792 |
| **5** | 1.0899 | 0.0587 | 1.0051 | 0.1335 |
| ***Exp*** | Qdot DMSO | | Qdot LatA | |
| *States* | D($\mu$m$^2$/s) | Prob | D($\mu$m$^2$/s) | Prob |
| **1** | $< 10^{-5}$ | 0.1469 | $< 10^{-5}$ | 0.1454 |
| **2** | 0.0101 | 0.3224 | 0.0090 | 0.2783 |
| **3** | 0.0376 | 0.3849 | 0.0361 | 0.3673 |
| **4** | 0.2230 | 0.1015 | 0.1619 | 0.1565 |
| **5** | 1.6021 | 0.0444 | 1.3978 | 0.0525 |

**Table 3.6:** Results of SPT-$\infty$ for experimental data on B cells, where IgM were labelled using either directly conjugated Fab anti-IgM (Cy3) or biotinylated anti-IgM plus avidin-conjugated Qdots. For these experiments, the cells were first treated with either DMSO control or LatA. We find that Cy3-LatA trajectories show faster state-wise diffusion coefficients compared to control. We find no meaningful difference between the Qdot DMSO and Qdot LatA experiments.

## 3.7 Discussion

In this chapter, we have described and implemented a novel iHMM algorithm for multiple state discrimination for SPT data analysis. We have also demonstrated its use on a collection of high-quality data that were previously analyzed using a two-state HMM. We believe that the iHMM approach is a rational choice for this longstanding state counting problem and is superior to alternative approaches such as the use of information criteria to distinguish among multiple models. The iHMM is a form of multimodel inference, across a set of possible models. If we were to perform a model comparison via an information criterion, we would first need to fit to some number of specific models, each of which would effectively be conditioned on its own correctness. It is not possible to compare the likelihoods of such individual models directly, and the uncertainty (variance in the parameter estimates) is also underestimated because our lack of knowledge about which model is correct is not included. The iHMM naturally incorporates the uncertainty due to model selection, while information criteria approaches are rather different and do not capture the uncertainty beyond a correction for the number of additional parameters in the more-complex model [3, 44]. Of course, information criteria have an important role to play when computational efficiency becomes important, as well as allowing model comparison when the possible models do not fit neatly together.

We were also able to improve the speed of convergence of the algorithm via a simple approach of merging nearby states using a criterion based on the Bhattacharya distance between distributions. This approach substantially improved the speed of the algorithm, although at the expense of causing some loss of accuracy in that not all runs converged to the correct number of states when using simulated data. We feel that the increase in speed, which allows many runs to be performed quite economically in comparison to a single use of the full algorithm, substantially outweighs the loss of confidence in any particular run.

We used our algorithm to re-examine SPT data obtained for BCRs moving on the surface of live B cells, using two different labelling strategies

and with pharmacological perturbation of the actin cytoskeleton. We found that BCRs can transition among four to five distinct diffusive states, with a wide range of diffusion coefficients. The slowest detected states appear to reflect transient immobility, while the fastest state probably reflects occasional tracking errors. Our results on the difference between Qdot-labelled molecules and Cy3-labelled molecules are in general agreement with our previous work [1], where we conclude that Qdot-labelling impairs molecule mobility. Moreover, our analysis confirms previous results of the actin cytoskeleton being an important regulator of receptor mobility. On LatA-treated cells, we observed higher mobility of the receptors, suggesting that disruption of actin cytoskeleton allows faster BCR motion.

In recent work, Rey-Suarez et al. [102] found that BCRs exist in 8 distinctive diffusive states. They used an expectation maximization approach based on a Gaussian mixture model, and allowing no transitions between states. In their work, the trajectories were split into segments each containing 15 frames. Each segment was assumed to arise from a single diffusive state. Similar to our findings here, they report a large number of diffusive states, confirming the heterogeneity and complexity of the cell membrane. However, since no transitions are allowed within each 15-frame segment, some segments could arise from a combination of diffusive states, yielding an overestimate of the optimal number of states. Nonetheless, the work of Rey-Suarez supports the use of multi-state models and shows their potential to support biological discovery.

Our approach to this problem has some potential weaknesses that we intend to address in future work. We have applied a restricted model of particle motion as a multi-state diffusion process. An alternative line of attack could allow for transient confinement of the particle within a potential well, or other forms of nondiffusive motion, as has previously been implemented by others in HMM and other settings[7, 60, 121]. Modified HMM models reflecting these kinds of additional complexity should be relatively simple to implement within the iHMM framework. Second, we did not account for localization errors in our current implementation. This is known to reduce the accuracy of estimation of diffusion coefficients in related settings [6], and

was incorporated into the recent study of Rey-Suarez et al. [102].

In summary, we have presented a generalizable approach that extends the potential applicability of HMMs for SPT data. More broadly, we have shown a novel application of the iHMM method, further proving that it is an excellent tool for quantification of experimental biophysics data [115]. All experimental datasets and software are freely available (https://github.com/rcardim/iHMMSPT).

# CHAPTER 4

## Infinite Hidden Markov Model with correction for measurement errors

As discussed in Chapter 2, the inclusion of experimental errors allows for a more accurate estimation of states and their parameters. There are two types of errors in the tracking of microscopic particles: the static error and the dynamical error. The static error is due to the background noise and is modelled as a usual Gaussian noise (white noise). The dynamical error is due to the *blur* effect, caused by the mobility of the particles while the snapshots of their configuration are taken.

In this Chapter, we further develop the infinite state hidden Markov model of the previous chapter to include these experimental errors. The Bayesian nonparametric framework is left unchanged. However, the emission distribution that models the relationship between the observed data and the parameters of the model is changed to properly take into account the experimental errors.

This chapter starts with a brief explanation of the measurement process, and how we build this process into the Bayesian nonparametric framework. After developing the new algorithm, and validating it with simulated data,

we discuss a way to improve the accuracy of the model. Finally, we discuss the results of the algorithm applied to experimental data and compare these to results obtained from these same datasets using our previous models.

## 4.1 Including experimental errors in the inference

In Chapter 2, we obtained the following distribution for $N$ displacements within a two state Markov model, where each states follows the measurement process.

$$P(\Delta_1 \cdots \Delta_N) \propto \frac{1}{\|Cov_{N\times N}\|} \exp\left( -\frac{1}{2} \left[\Delta_1 \cdots \Delta_N\right] Cov^{-1}_{N\times N} \begin{bmatrix} \Delta_1 \\ \vdots \\ \Delta_N \end{bmatrix} \right), \quad (4.1)$$

where $\mathrm{Cov}_{N\times N} =$

$$\begin{cases} \langle \Delta_i \Delta_j \rangle = 2D\left((i-1)\,\tau \to i\tau\right)\frac{\tau}{3} + 2D\left(i\tau \to (i+1)\,\tau\right)\frac{\tau}{3} + 2\sigma^2, \text{ for } j = i, \\ \langle \Delta_i \Delta_j \rangle = 2D\left(i\tau \to (i+1)\,\tau\right)\frac{\tau}{6} - \sigma^2, \text{ for } j = i-1 \text{ or } j = i+1, \\ 0, \text{ otherwise.} \end{cases} \quad (4.2)$$

is the covariance matrix of the process, and the diffusion coefficients vary with the hidden state at each displacement:

$$D\left(i\tau \to (i+1)\,\tau\right) = \begin{cases} D_1, \text{ for } s_i = 1 \\ D_2, \text{ for } s_i = 2, \end{cases}$$

for $i = 1, \ldots, N$.

The dynamical error (blur) and the static error are included in the covariance elements. The static error appears explicitly in the matrix elements through its standard deviation $\sigma$. The dynamical error is responsible for the constants multiplying the diffusivities. These terms were previously calculated in Chapter 2, subsection 2.1.3, through the calculation of the variables

$R_1$ and $R_2$ for an uniform illuminating profile, i.e. the shutter function is equal to $1/\tau$.

From this equation, we observe that the distributions and covariance matrix do not depend on the number of states but only on the diffusivities. Thus, we can generalize this model to a $K$ states model, and update the diffusion coefficients to:

$$D\left(i\tau \to (i+1)\,\tau\right) = \begin{cases} D_1, \text{ for } s_i = 1 \\ D_2, \text{ for } s_i = 2, \\ \cdots, \cdots, \\ D_k, \text{ for } s_i = k, \\ \cdots, \cdots, \\ D_K, \text{ for } s_i = K, \end{cases},$$

for $i = 1, \ldots, N$.

This new process provides a generalization of the number of states $K$, allowing it to be any integer. The combination of this process with the infinite Hidden Markov model framework yields an algorithm that estimates the number of states, their diffusion coefficients, and their transition probabilities while taking into account the experimental errors. Thus, we need to adapt the SPT-$\infty$ algorithm developed in Chapter 3 so that it considers the new measurement process, instead of a simple diffusion process. Two parts of the algorithm are affected by this change: the emission model, and the sampling of the state sequence. We explain these changes to the algorithm in the next sections.

### 4.1.1 Sampling state chain

In the SPT-$\infty$, the sampling of the state trajectory is performed through the beam sampler. As explained in Chapter 3, this sampler works iteratively as in Equation (3.18) through the Markov property of $p(s_t \mid s_1, \ldots, s_{t-1}) = p(s_t \mid s_{t-1})$. For this measurement process, the likelihood of the data given the parameters and the sequence of states is a multivariate Gaussian as

in Equation (4.1). Therefore, all the displacements are correlated and the Markov assumption is violated. Because of that, we make an extra assumption on the measurement process, so that it satisfies the Markov property and keeps the correlation among the displacements. We assume that transitions between states can only occur after two frames. In other words, we paired the displacements, and assume that displacements in each pair occupy the same Markovian state. Moreover, we only consider the correlation within each pair of displacements. This assumption might not hold if the transition probability between states is high. Thus, the likelihood is essentially given by multiplication of multivariate two-dimensional Gaussians:

$$P(\Delta_1 \cdots \Delta_N \mid \mathrm{Cov}_{2i-1,2i}) \propto$$
$$\prod_{i=1}^{\leq \frac{N}{2}} \frac{1}{\|\mathrm{Cov}_{2i-1,2i}\|} \exp\left(-\tfrac{1}{2}\begin{bmatrix}\Delta_i \Delta_{i+1}\end{bmatrix} \mathrm{Cov}_{2i-1,2i}^{-1} \begin{bmatrix}\Delta_{2i-1} \\ \Delta_{2i}\end{bmatrix}\right), \tag{4.3}$$

where

$$\mathrm{Cov}_{2i-1,2i} = \begin{bmatrix} \langle \Delta_{i,i} \rangle & \langle \Delta_{i,i+1} \rangle \\ \langle \Delta_{i+1,i} \rangle & \langle \Delta_{i+1,i+1} \rangle \end{bmatrix},$$

with

$$\langle \Delta_{i,i} \rangle = 4D_{s_i}\frac{\tau}{3} + 2\sigma^2, \tag{4.4}$$
$$\langle \Delta_i \Delta_{i+1} \rangle = D_{s_i}\frac{\tau}{3} - \sigma^2. \tag{4.5}$$

In summary, we are hypothesizing that the parameters that maximize the full likelihood for $N$ displacements ($N \times N$ tridiagonal covariance matrix) can be approximated by the parameters that maximize the likelihood built by $N/2$ disjoint blocks ($2 \times 2$) of the full covariance matrix. This approximation should hold for large $N$, giving that it basically discards information from the data on the correlations between the last displacement in a pair with the first displacement in the next pair.

### 4.1.2 Emission Model and Conjugate Prior

From this emission model, we can calculate the posterior distribution if we use a prior distribution that is the conjugate to the likelihood. It is known that the Wishart distribution is the conjugate prior distribution to the multivariate Gaussian distribution with a known mean [36]. The Wishart distribution is a generalization of the Gamma distribution to multiple dimensions [140]. The probability density function of a symmetric matrix $M$ of size $p \times p$ distributed according to a Wishart distribution is given by:

$$W(\Psi, \nu) = \frac{1}{2^{\nu p/2}} \|\Psi\|^{\nu/2} \Gamma_p \left(\frac{\nu}{2}\right) |M|^{(\nu-p-1)/2} e^{\left(-\frac{1}{2}\mathrm{tr}\left(\Psi^{-1}M\right)\right)}, \tag{4.6}$$

where $\Psi$ and $\nu$ are the parameters of the Wishart distribution. $V$ is a symmetric positive definite $p \times p$ matrix, $\nu$ is the degrees of freedom, and $\Gamma_p$ is the multivariate Gamma function.

Since in our algorithm we sample the covariance matrix, we must use the inverse Wishart distribution as a conjugate prior.

The likelihood of a pair of displacements is given by

$$P\left(\Delta_{2i-1}, \Delta_{2i} \mid Cov_{2i-1,2i}\right) =$$
$$\frac{1}{\|Cov_{2i-1,2i}\|} \exp\left(-\frac{1}{2}\left[\Delta_{2i-1}\Delta_{2i}\right] Cov_{2i-1,2i}^{-1} \begin{bmatrix}\Delta_{2i-1}\\ \Delta_{2i}\end{bmatrix}\right).$$

The inverse Wishart prior of the covariance matrix is given by:

$$P(Cov \mid \Psi, \nu) = \frac{\mid \Psi \mid^{\nu/2}}{2^{\nu}\Gamma_2\left(\frac{\nu}{2}\right)} \mid Cov \mid^{(\nu+3)/2} e^{-\frac{1}{2}tr\left(\Psi Cov^{-1}\right)},$$

where $\Psi$ and $\nu$ are hyperparameters for the inverse Wishart distribution. $\Psi$ is a matrix of the same size as the random variable. We use the identity matrix of size $2 \times 2$ for $\Psi$. $\nu$ is the degree of freedom parameter, and for our algorithm, we fix $\nu = 3$. The fewer the degrees of freedom $\nu$, the larger is the variability of the samples. Then, the least informative prior is $\nu = p$. We also set our $\Psi = I$, where $I$ is the identity matrix. Then, the posterior

is given by:

$$P(Cov \mid \Delta_{2i-1}, \Delta_{2i}) = \frac{\mid \Psi_{pos} \mid^{\nu_{pos}/2}}{2^{\nu_{pos}}\Gamma_2\left(\frac{\nu_{pos}}{2}\right)} \mid Cov \mid^{(\nu_{pos}+3)/2} e^{-\frac{1}{2}tr\left(\Psi_{pos}Cov^{-1}\right)}, \tag{4.7}$$

where the posterior parameters $\nu_{pos}$ and $\Psi_{pos}$ are defined by:

$$\begin{aligned} \Psi_{pos} =& \left(\Psi + \sum_{1}^{\leq N/2} \begin{bmatrix} \Delta_{2i-1} \\ \Delta_{2i} \end{bmatrix} \begin{bmatrix} \Delta_{2i-1}\Delta_{2i} \end{bmatrix}\right), \\ \nu_{pos} =& \nu + N/2. \end{aligned}$$

### 4.1.3 Modified Algorithm

The steps of the infinite hidden Markov model with correction for positional correlations due to experimental errors are as follows. This algorithm should be compared with the SPT-$\infty$ algorithm, see Chapter 3, page 85. Throughout this thesis, we refer to the infinite hidden Markov model with correction for positional correlations due to experimental errors as SPT-$\infty$E.

A Generate an auxiliary variable for each time point of the dataset, i.e. $\boldsymbol{u}^{(l+1)} = (u_1, \ldots, u_N)$ using $u_t \sim \mathcal{U}\left(0, \pi_{s_{t-1} \to s_t}\right)$.

B Use $\boldsymbol{u}^{(l+1)}$ to expand the state space $\mathbb{S}$.

C Generate a new sequence of states $\boldsymbol{s}^{(l+1)}$ for each pair of displacements using a modified version of the forward-filtering backward algorithm, whose likelihood is defined by equation (4.3).

D Compress the state space $\mathbb{S}$, excluding the states that are not visited in $\boldsymbol{s}^{(l+1)}$.

E Generate $\widetilde{Cov}^{(l+1)}$ using the posterior distribution given by (4.7).

F Generate $\tilde{\beta}^{(l+1)}$ using the new state sequence.

G Generate the transition probability matrix $\tilde{\tilde{\pi}}^{(l+1)}$ using $\tilde{\beta}^{(l+1)}$ and $\boldsymbol{s}^{(l+1)}$.

A brief note on the notation: $l$ is the iteration index of the algorithm, variables that are defined over the experimental time-space are bold, and variables that are vectors, matrices or tensors over the state space have a tilde or double tilde.

## 4.2 Validation

The first step for validation of this algorithm is to check whether the algorithm converges to the correct number of states for the simplest case: when the experimental error is only due to static errors. We simulate displacements of particles undergoing the measurement process with only a static error of standard deviation 0.001 in an appropriate arbitrary unit, for different numbers of diffusive states. Table 4.1 shows some of the parameters for each dataset. The difference among the diffusion coefficients for each state goes from 10-fold to 4-fold. For each dataset, we simulated 1 particle with 20000 displacements.

| Number of States $K$ | $\sigma$ | $D_1$ | $D_2$ | $D_3$ | $D_4$ |
|---|---|---|---|---|---|
| 2 states | 0.01 | 0.01 | 0.1 | —— | —— |
| 3 states | 0.01 | 0.01 | 0.1 | 0.5 | —— |
| 4 states | 0.01 | 0.01 | 0.1 | 0.5 | 2 |

**Table 4.1:** Parameters used for the simulation of trajectories under multi-state Brownian diffusion with only static error. $D_1$, $D_2$, $D_3$, and $D_4$ are the diffusivities of each state. $\sigma$ is the standard deviation of the static error. All quantities have appropriate arbitrary units.

The transition matrix for simulated dataset of trajectories is on Table 4.2.

| K | True T | | | |
|---|---|---|---|---|
| 2 | 0.8 | 0.2 | | |
| | 0.2 | 0.8 | | |
| K | True T | | | |
| 3 | 0.8 | 0.1 | 0.1 | |
| | 0.1 | 0.8 | 0.1 | |
| | 0.1 | 0.1 | 0.8 | |
| K | True T | | | |
| 4 | 0.8 | 0.1 | 0.05 | 0.05 |
| | 0.1 | 0.8 | 0.05 | 0.05 |
| | 0.05 | 0.05 | 0.8 | 0.1 |
| | 0.05 | 0.05 | 0.1 | 0.8 |

**Table 4.2:** Transition matrices used for the simulation of trajectories under multi-state Brownian diffusion.

First, the algorithm completes 10000 iterations as the burn-in phase. Next, 5000 iterations are performed to generate a distribution for each parameter of each state. Figure 4.1 shows the number of states for each dataset for 10 chains. For the case of $K = 2$, all chains achieve the correct number of states. For $K = 3$ and $K = 4$ states, $\frac{9}{10}$ of the chains converge to the correct value.

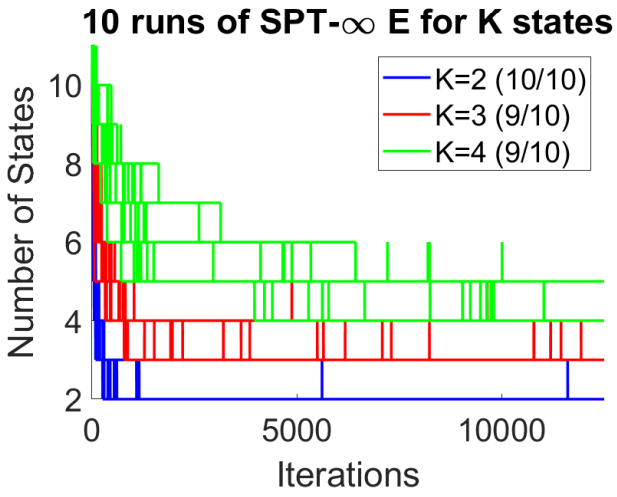



**Figure 4.1:** Number of states estimates for simulated dataset with only static error ($\sigma = 0.001$ with appropriate unit) using the SPT-$\infty$E algorithm. Ten different runs are shown for each set of simulated data (with colours indicated). Convergence was generally achieved after about 10000 iterations. 9 out of 10 runs converged to the correct number of states for the four- and three-state dataset, and 10 out of 10 runs converged correctly for the two-state dataset.

We also validated the algorithm for simulated data with both static and dynamical errors. Figure 4.2 shows the results for one validation set. Again, we simulated 1 particle over 20000 frames, with parameters as in Table 4.1 but with an additional dynamical error. We assumed that the shutter function is uniform, and therefore, $R_1 = \frac{1}{3}$, and $R_2 = \frac{1}{2}$ in Equations (2.1) and (2.8). During the 10000 iterations, the algorithm did not converge to the correct number of states for any of the 4 states dataset's chains. The algorithm arrives at the correct number of states 2 times out of 10 times for the 3 states dataset and all 10 times for the 2-state dataset. Thus, the performance of the algorithm is not ideal. We see the need for a higher number of iterations and/or more data so that convergence to the correct number of states is achieved. However, increasing the number of iterations or amount of data yields an increase in computational time and memory

allocation. Here, we propose an additional step to the algorithm to tackle this issue.

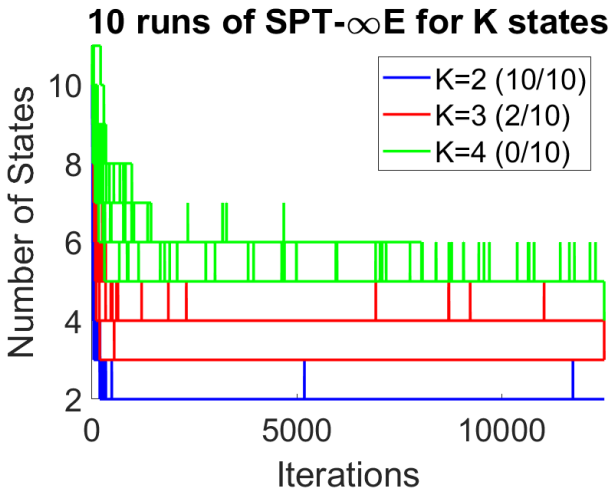



**Figure 4.2:** Number of states estimates for each simulated dataset with both static error ($\sigma = 0.001$) and dynamic error (uniform illuminating profile) using the SPT-$\infty$E algorithm. Ten different runs are shown for each set of simulated data (with colours indicated). Convergence was generally achieved after about 10000 iterations. For the 4 states dataset, no chain achieved the correct number of states, 2 out of 10 runs converged to the correct number of states for the three-states dataset, and 10 out of 10 runs converged correctly for the two-states dataset.

The estimated values of the covariance matrix for each state provide further explanation of the algorithm performance. Below is the true covariance matrix for each state of the dynamic dataset, and the estimated covariance matrices for each state of one of the 10 chains, for 3- and 4-state models.

**3-state True Value:**

$$Cov_1 = \begin{bmatrix} 0.4060 & 0.1000 \\ 0.1000 & 0.4060 \end{bmatrix} \times 10^{-3},\ Cov_2 = \begin{bmatrix} 0.0040 & 0.0010 \\ 0.0010 & 0.0040 \end{bmatrix},$$

$$Cov_3 = \begin{bmatrix} 0.0202 & 0.0050 \\ 0.0050 & 0.0202 \end{bmatrix}$$

**3-state Estimated Value:**

$$Cov_1 = \begin{bmatrix} 0.9916 & 0.1755 \\ 0.1755 & 0.9915 \end{bmatrix} \times 10^{-3},\ Cov_2 = \begin{bmatrix} 0.0238 & 0.0079 \\ 0.0079 & 0.0238 \end{bmatrix},$$

$$Cov_3 = \begin{bmatrix} 0.0073 & 0.0018 \\ 0.0018 & 0.0073 \end{bmatrix},\ Cov_4 = \begin{bmatrix} 0.0222 & 0.0057 \\ 0.0057 & 0.0222 \end{bmatrix},$$

**4-state True Value:**

$$Cov_1 = \begin{bmatrix} 0.4060 & 0.1000 \\ 0.1000 & 0.4060 \end{bmatrix} \times 10^{-3},\ Cov_2 = \begin{bmatrix} 0.0040 & 0.0010 \\ 0.0010 & 0.0040 \end{bmatrix},$$

$$Cov_3 = \begin{bmatrix} 0.0202 & 0.0050 \\ 0.0050 & 0.0202 \end{bmatrix},\ Cov_4 = \begin{bmatrix} 0.0808 & 0.0202 \\ 0.0202 & 0.0808 \end{bmatrix}$$

**4-state Estimated Value:**

$$Cov_1 = \begin{bmatrix} 0.0014 & 0.0002 \\ 0.0002 & 0.0014 \end{bmatrix},\ Cov_2 = \begin{bmatrix} 0.0131 & 0.0031 \\ 0.0031 & 0.0131 \end{bmatrix},$$

$$Cov_3 = \begin{bmatrix} 0.0459 & 0.0117 \\ 0.0117 & 0.0459 \end{bmatrix},\ Cov_4 = \begin{bmatrix} 0.0481 & 0.0105 \\ 0.0105 & 0.0481 \end{bmatrix},$$

$$Cov_5 = \begin{bmatrix} 0.0845 & 0.0232 \\ 0.0232 & 0.0845 \end{bmatrix}$$

As one can see, some of the estimated covariance matrices for the different states of the dataset with both static and dynamic error are close to their true values. For example, each estimated value for the 3-state case has a similar value to one of the true covariance matrices for state 3. More importantly, the extra state (state 4) has an estimated covariance matrix similar to state 2, suggesting that these two states are not different and that they will eventually merge as we increase the number of iterations. Thus, instead of running more iterations until they merge, we add a post-processing

step to the algorithm. This post-processing step analyzes the distribution of the covariance matrix for each state generated by the burn-in phase. If any two states have a similar covariance matrix, the states are merged. After merging the appropriate states, we re-initiated the algorithm with new values for the number of states, covariance matrices, transition matrices, and so on. Next, the algorithm iterates for some number $m$ of steps. The final result is then the $m$ size sample of the distribution of the parameter, for each state. This step is similar to the condition for merging states from the SPT-$\infty$ algorithm developed in Chapter 3. There, the condition is based on the Bhattacharya distance between the Gaussian distributions of each state, and it is applied during the whole *burn-in* phase of the MCMC algorithm. For the SPT-$\infty$E algorithm, the merging step happens only once immediately after the *burn-in* phase as explained below.

The 4-state case has one estimated covariance matrix that is not similar to any of the true values. The size of the dataset is the main reason for that. Once we simulate more data points, and run the algorithm with more displacements, the estimated covariance matrices are close to the correct value in all examined cases.

In the next section, the merging process is described in more detail, the validation of the algorithm is provided, and the validation is repeated using a larger dataset.

## 4.3 Merging states to accelerate convergence

Within this method to improve convergence, the decision to merge states is based on the proximity of the elements of the state covariance matrices. Let us use as an example the estimated covariance matrices for the 3-state data shown above. The estimated covariance for state 2 is

$$\mathrm{Cov}_2 = \begin{pmatrix} 0.0238 & 0.0079 \\ 0.0079 & 0.0238 \end{pmatrix}.$$

This matrix is close to the estimated covariance for state 4,

$$\mathrm{Cov}_4 = \left(\begin{smallmatrix} 0.0222 & 0.0057 \\ 0.0057 & 0.0222 \end{smallmatrix}\right).$$

The absolute difference between these matrices is equal to:

$$\begin{bmatrix} 0.0016 & 0.0023 \\ 0.0023 & 0.0016 \end{bmatrix} \tag{4.8}$$

The main diagonal of (4.8) represents around 7% of the value of the estimated covariance matrix. We obtain a small difference, however how one could define the closeness among different covariance matrices? Here, we first define a metric for the distance between two matrices.

Let $A$, and $B$ be $2 \times 2$ matrices, and define

$$d(A, B) = \frac{\sum_{i,j} \mid A_{i,j} - B_{i,j} \mid}{\sum_{i,j} (A_{i,j} + B_{i,j})} \tag{4.9}$$

We will merge states when their distance is smaller than a threshold, i.e. if $d(A, B) \leq h$, then $A$ is merged into $B$. We can calculate a useful value for this threshold by considering the ratio between the diffusion coefficient of each covariance matrix. Let $D_A$, and $D_B$, be the diffusion coefficients of states generating covariance matrices $A$ and $B$. We have:

$$A = \begin{bmatrix} \frac{4D_A\tau}{3} + 2\sigma^2 & \frac{2D_A\tau}{6} - \sigma^2 \\ \frac{2D_A\tau}{6} - \sigma^2 & \frac{4D_A\tau}{3} + 2\sigma^2 \end{bmatrix} \tag{4.10}$$

and

$$B = \begin{bmatrix} \frac{4D_B\tau}{3} + 2\sigma^2 & \frac{2D_B\tau}{6} - \sigma^2 \\ \frac{2D_B\tau}{6} - \sigma^2 & \frac{4D_B\tau}{3} + 2\sigma^2 \end{bmatrix} \tag{4.11}$$

Then, $d(A, B)$ is given by

$$d(A,B) = \frac{\frac{8\tau|D_A - D_B|}{3} + \frac{4\tau|D_A - D_B|}{6}}{\frac{8\tau(D_A + D_B)}{3} + \frac{4\tau(D_A + D_B)}{6} + 4\sigma^2} \tag{4.12}$$

Let $D_A = rD_B$, where $r \geq 1$ without losing generality to obtain

$$\begin{aligned} d(A,B) &= \frac{D_B \frac{20\tau(r-1)}{6}}{D_B \frac{20\tau(r+1)}{6} + 4\sigma^2} \\ d(A,B) &\leq \frac{D_B \frac{20\tau(r-1)}{6}}{D_B \frac{20\tau(r+1)}{6}} \\ d(A,B) &\leq \frac{(r-1)}{(r+1)} \end{aligned} \tag{4.13}$$

For a chain with a number of states $K$, we define the threshold $h$ as $h = \frac{a-1}{a+1}$ where $a$ is the infimum of the ratios among the diffusion coefficients, $a = \inf\{r_i \mid r_i = \frac{D_i}{D_j}, i, j = 1, \ldots, K, i \neq j\}$. Therefore, for every pair of states whose covariance matrices' distance is smaller than $h$, the states are merged. Then, the new algorithm is:

A Generate auxiliary variable for each time point of the dataset, i.e. $\boldsymbol{u}^{(l+1)} = (u_1, \ldots, u_N)$ using $u_t \sim \mathcal{U}\left(0, \pi_{s_{t-1} \to s_t}\right)$.

B Use $\boldsymbol{u}^{(l+1)}$ to expand the state space $\mathbb{S}$.

C Generate new sequence of states $\boldsymbol{s}^{(l+1)}$ for each pair of displacements using a modified version of the forward-filtering backward algorithm, whose likelihood is by equation (4.3).

D Compress the state space $\mathbb{S}$, excluding the states that are not visited in $\boldsymbol{s}^{(l+1)}$.

E Generate $\tilde{Cov}^{(l+1)}$ using the posterior distribution given by (4.7).

F Generate $\tilde{\beta}^{(l+1)}$ using the new state sequence.

G Generate the transition probability matrix $\tilde{\tilde{\pi}}^{(l+1)}$ using $\tilde{\beta}^{(l+1)}$ and $\boldsymbol{s}^{(l+1)}$.

H if $l =$ burn-in, calculate the average covariance matrix for every state, and merge them when their average covariance matrices distance is smaller than a threshold $d(Cov_i, Cov_j) < h$. After merging, we restart the algorithm with the new merged states.

To validate this approach for merging states that are not substantially different from each other, we ran the algorithm with this additional compression step on the simulated data described by Table 4.1. Figure 4.3 shows results from 10 chains for each dataset with different states and with only static error, and Figure 4.4 reports the results for the second simulated dataset with both static and dynamical errors.

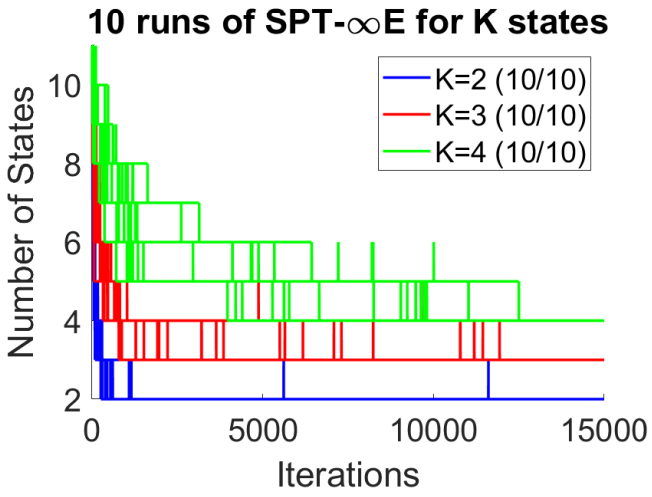



**Figure 4.3:** Estimation of number of states at each iteration for each simulated dataset with only static error ($\sigma = 0.001$) using the SPT-$\infty$E algorithm with the merging states step. Ten different runs of the algorithm are shown for each set of simulated data (with colours indicated). Convergence was achieved after about 10000 iterations. All runs converged to the correct number of states for all datasets.

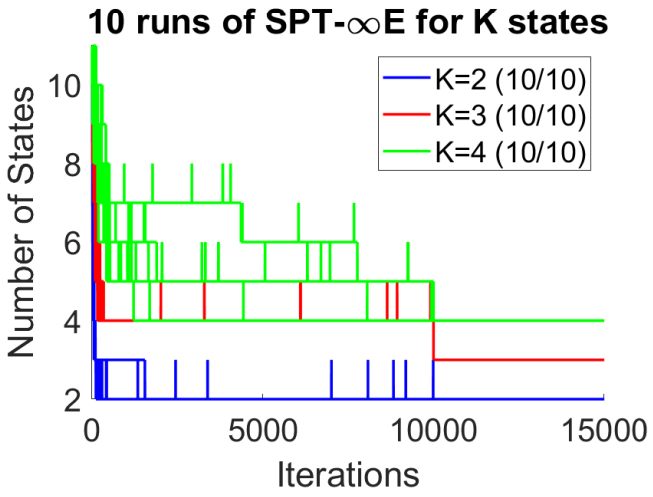



**Figure 4.4:** Estimation of number of states at each iteration for each simulated dataset with both static error ($\sigma = 0.001$) and dynamical error using the SPT-$\infty$E algorithm with the merging states step. Ten different runs of the algorithm are shown for each set of simulated data (with colours indicated). Convergence was achieved after about 10000 iterations. All runs converged to the correct number of states for all datasets.

To further validate the algorithm, we analyze the estimated covariance matrix for each state. Below, we have the true value of each state's covariance matrix and each estimated covariance matrix for one of the 10 runs.

For the 3-state dataset:

**True Value:**

$$Cov_1 = \begin{bmatrix} 0.4060 & 0.1000 \\ 0.1000 & 0.4060 \end{bmatrix} \times 10^{-3},\ Cov_2 = \begin{bmatrix} 0.0040 & 0.0010 \\ 0.0010 & 0.0040 \end{bmatrix},$$

$$Cov_3 = \begin{bmatrix} 0.0202 & 0.0050 \\ 0.0050 & 0.0202 \end{bmatrix}$$

**Estimated:**

$$Cov_1 = \begin{bmatrix} 0.9998 & 0.1951 \\ 0.1951 & 0.9788 \end{bmatrix} \times 10^{-3},\ Cov_2 = \begin{bmatrix} 0.0073 & 0.0018 \\ 0.0018 & 0.0073 \end{bmatrix},$$

$$Cov_3 = \begin{bmatrix} 0.0220 & 0.0063 \\ 0.0063 & 0.0224 \end{bmatrix},$$

For the 4-state dataset:

**True Value:**

$$Cov_1 = \begin{bmatrix} 0.4060 & 0.1000 \\ 0.1000 & 0.4060 \end{bmatrix} \times 10^{-3},\ Cov_2 = \begin{bmatrix} 0.0040 & 0.0010 \\ 0.0010 & 0.0040 \end{bmatrix},$$

$$Cov_3 = \begin{bmatrix} 0.0202 & 0.0050 \\ 0.0050 & 0.0202 \end{bmatrix},\ Cov_4 = \begin{bmatrix} 0.0808 & 0.0202 \\ 0.0202 & 0.0808 \end{bmatrix}$$

**Estimated:**

$$Cov_1 = \begin{bmatrix} 0.0013 & 0.0002 \\ 0.0002 & 0.0014 \end{bmatrix},\ Cov_2 = \begin{bmatrix} 0.0134 & 0.0033 \\ 0.0033 & 0.0134 \end{bmatrix},$$

$$Cov_3 = \begin{bmatrix} 0.0454 & 0.0108 \\ 0.0108 & 0.0463 \end{bmatrix},\ Cov_4 = \begin{bmatrix} 0.0820 & 0.0226 \\ 0.0226 & 0.0825 \end{bmatrix},$$

We obtained results that were in good agreement with the true values for the 2-state and 3-state datasets. However, for the 4-state dataset, even though the correct number of states was returned for all runs, the covariance matrices do not agree with the true values for each state. The reason for that is the size of the dataset. Here we used a dataset of a single particle trajectory with 20000 frames, but this is not enough to accurately predict the covariance matrix for 4 states.

To confirm that larger datasets would allow us to obtain accurate covariance matrices for 4 states, we simulated a new dataset, with the same diffusion coefficients and transition matrix as before. However, we simulated 100 particles, each one with 300 frames, yielding a total of 30000 frames. This dataset is not only larger but similar to datasets acquired from experiments since we have many particles with a small number of frames. Figure 4.5 shows the state convergence results for this dataset with both static and dynamical error.

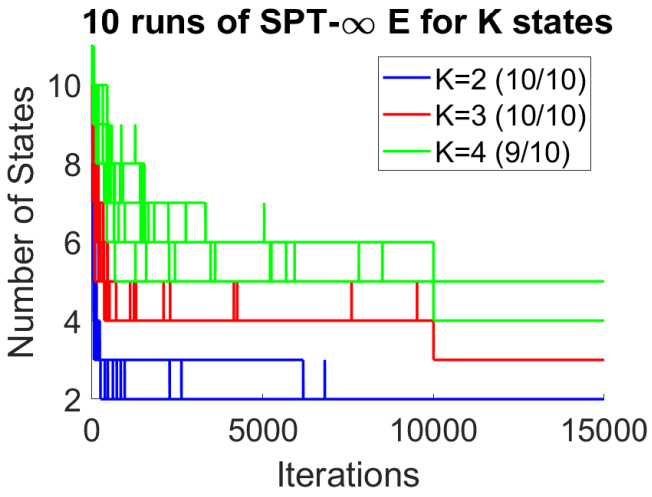



**Figure 4.5:** Estimation of number of states at each iteration for each simulated dataset with both static error ($\sigma = 0.001$) and dynamical error using the SPT-$\infty$E algorithm with the merging states step for a dataset with 30000 frames. Ten different runs of the algorithm are shown for each set of simulated data (with colours indicated). Convergence was generally achieved after about 10000 iterations. 9 out of 10 runs converged to the correct number of states for the four-state dataset, and 10 out of 10 runs converged correctly for both three-state and two-state datasets.

For both 2-state and 3-state datasets, all runs converged to the correct number of states. For the 4-state dataset, 9/10 of the runs converged to the correct number of states. Below, one can find the results for each covariance matrix of each state in one of the 10 runs. The new results are much more accurate than before, showing that the size of the dataset was the culprit for the poor estimation of the 4-state dataset covariance matrices.

For the 2-state dataset, the difference between the true covariance matrix and the estimated covariance matrix $\|Cov_i^{true} - Cov_i^{est}\|$ for each state $i$ is:

$$\|Cov_1^{true} - Cov_1^{est}\| = \begin{bmatrix} 3 & 0.2 \\ 0.2 & 3 \end{bmatrix} \times 10^{-4},$$

$$\|Cov_2^{true} - Cov_2^{est}\| = \begin{bmatrix} 1 & 0.05 \\ 0.05 & 1 \end{bmatrix} \times 10^{-4}.$$

For the 3-state dataset $\|Cov_i^{true} - Cov_i^{est}\|$ for each state $i$ is:

$$\|Cov_1^{true} - Cov_1^{est}\| = \begin{bmatrix} 4 & 0.6 \\ 0.6 & 4 \end{bmatrix} \times 10^{-4},$$

$$\|Cov_2^{true} - Cov_2^{est}\| = \begin{bmatrix} 3 & 0.6 \\ 0.6 & 3 \end{bmatrix} \times 10^{-3},$$

$$\|Cov_3^{true} - Cov_3^{est}\| = \begin{bmatrix} 2 & 0.4 \\ 0.4 & 2 \end{bmatrix} \times 10^{-3}$$

For the 4-state dataset$\|Cov_i^{true} - Cov_i^{est}\|$ for each state $i$ is:

$$\|Cov_1^{true} - Cov_1^{est}\| = \begin{bmatrix} 5 & 0.5 \\ 0.5 & 5 \end{bmatrix} \times 10^{-4},$$

$$\|Cov_2^{true} - Cov_2^{est}\| = \begin{bmatrix} 2 & 0.2 \\ 0.2 & 2 \end{bmatrix} \times 10^{-3},$$

$$\|Cov_3^{true} - Cov_3^{est}\| = \begin{bmatrix} 7 & 0.7 \\ 0.7 & 7 \end{bmatrix} \times 10^{-3},$$

$$\|Cov_4^{true} - Cov_4^{est}\| = \begin{bmatrix} 1 & 2 \\ 2 & 0.6 \end{bmatrix} \times 10^{-3}$$

Throughout the rest of this thesis, SPT-$\infty$E refers to the final version of the algorithm including the merging states step.

Finally, Figure 4.6 provides graphs with the values of the covariance matrix Cov for each iteration of the SPT-$\infty$E algorithm for the 3-state dataset. These graphs depicts a more detailed view of the convergence of the algorithm. As we can see, around the 10000th step, the algorithm converges, and the values for each element no longer vary much.

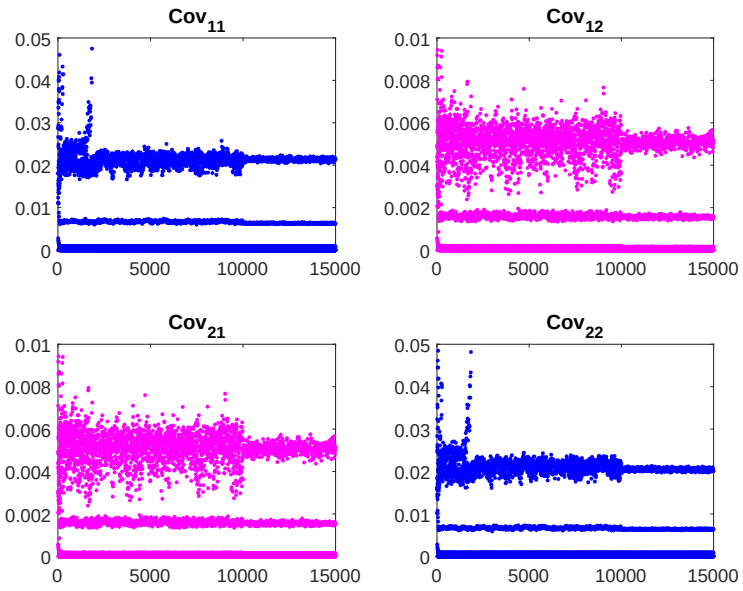



**Figure 4.6:** Estimates for each element of the Cov matrices for each iteration of the SPT-$\infty$E algorithm applied to the simulated dataset with 3 states. This is for one run of the simulated data with both errors, and with 30000 frames. We can observe the convergence of the algorithm in this plot. At first, there are many states, until the number of states converges for 3 states, and the values for each of the Cov elements converged to 3 values.

### Transition Matrix Estimates

The SPT-$\infty$E algorithm also estimates the transition matrix of the Markov process. We report the final estimates of the transition matrix for one run of each dataset, in Table 4.3.

We find that the transition matrices are reasonably recovered with good accuracy. Moreover, Figure 4.7 shows heatmaps of the average absolute difference between each estimated transition matrix and their respective true value, for all runs of the algorithm on each $k$-states dataset. Each heatmap reports $\overline{|T_{est} - T|}_{runs}$, where $T_{est}$ is the estimated transition matrix for a run

| K | True T | | | Estimated T | | | |
|---|---|---|---|---|---|---|---|
| 2 | 0.8 | 0.2 | | 0.7568 | 0.2432 | | |
| | 0.2 | 0.8 | | 0.2583 | 0.7417 | | |
| K | True T | | | Estimated T | | | |
| 3 | 0.8 | 0.1 | 0.1 | 0.7030 | 0.2172 | 0.0797 | |
| | 0.1 | 0.8 | 0.1 | 0.2058 | 0.7011 | 0.0930 | |
| | 0.1 | 0.1 | 0.8 | 0.1049 | 0.1590 | 0.7360 | |
| K | True T | | | | Estimated T | | | |
| 4 | 0.8 | 0.1 | 0.05 | 0.05 | 0.7135 | 0.1547 | 0.0999 | 0.0318 |
| | 0.1 | 0.8 | 0.05 | 0.05 | 0.1815 | 0.6645 | 0.1118 | 0.0421 |
| | 0.05 | 0.05 | 0.8 | 0.1 | 0.1039 | 0.1148 | 0.6013 | 0.1799 |
| | 0.05 | 0.05 | 0.1 | 0.8 | 0.0442 | 0.0382 | 0.2335 | 0.6840 |

**Table 4.3:** Estimated transition matrices for one of the 10 runs of SPT-$\infty$E over the three $K$-state datasets.

over one of the $K$-states dataset, and $T$ is the true value of the transition matrix for that dataset, and we averaged over all runs that converged to the correct number of states.

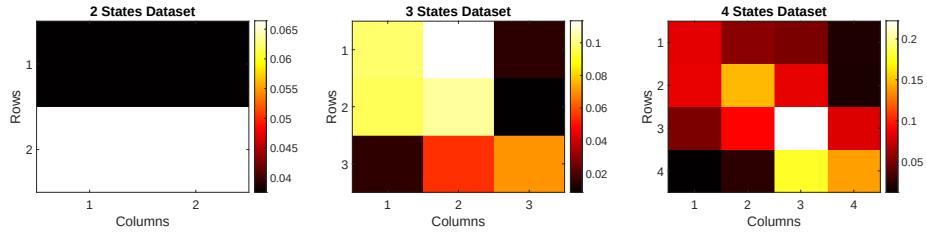



**Figure 4.7:** Average of absolute difference between true transition matrix and estimated transition matrix, for all runs that converge to the correct number of states.

## 4.4 Application to B cell receptor tracking data

Following the same procedure as in Chapter 2 and 3, we also apply the SPT-$\infty$E to experimental data. Again, we analyzed datasets from our work in [1]. This work investigates if two fluorescent labelling methods achieve similar results for the mobility of the labelled proteins. The first label is

a small organic fluorophore (Cy3), and the second labelling method uses quantum dots as the fluorescent probe. The advantage of quantum dots is that they do not photobleach, allowing for longer trajectories. However, the size of quantum dots is of the same order as the tracked molecules, which affects their mobility.

We examined six datasets. Three of these datasets are from Cy3-labelling experiments, while the other three are from Qdot-labelling experiments. The experiments were performed on three different days: one pair of Qdot and Cy3 labelling of the same batch of cells on each day. First, we generated only one chain with 15000 iterations for each dataset. From this, we obtained the following number of states for each dataset:

| Dataset | Cy3 1 | Cy3 2 | Cy3 3 | Qdot 1 | Qdot 2 | Qdot 3 |
|---|---|---|---|---|---|---|
| Number of States | 4 | 4 | 3 | 4 | 4 | 3 |

**Table 4.4:** Estimated number of states from SPT-$\infty$E following 150000 iterations.

We estimated four states for both labelling methods for the first and second experiments and three states for the third experiment. The covariance matrix estimates for each state of each of these estimations are shown on Tables 4.5 and 4.6. For the first experiment, two estimated covariance matrices are too similar for the Cy3-labelling method, even though we assumed a 2-fold difference between the diffusivities of each state in the merging process. The results contain some similarities and consistency for the three independent experiments. For example, the first state covariance matrix is very similar for a given labelling method for all three experiments, and the same was true for the second state. The discrepancies appear in the third state. For example, since there is no fourth state in the third experiment, it seems the third state for this experiment includes all of the trajectories with large diffusivity.

For the Qdot-labelling method, the first thing one observes is that the values of the covariance matrix diagonal for the first state are of the order of

| Datasets | Cy3 1 ($\mu m$) | | Cy3 2 ($\mu m$) | | Cy3 3 ($\mu m$) | |
|---|---|---|---|---|---|---|
| $\sigma_i$ | Covariance Matrix | | | | | |
| 1 | 0.0016 | -0.0005 | 0.0013 | -0.0004 | 0.0014 | -0.0004 |
| | -0.0005 | 0.0015 | -0.0004 | 0.0013 | -0.0004 | 0.0013 |
| 2 | 0.0083 | -0.0007 | 0.0064 | -0.001 | 0.009 | -0.0008 |
| | -0.0007 | 0.0085 | -0.001 | 0.0065 | -0.0008 | 0.0093 |
| 3 | 0.0400 | -0.0034 | 0.0229 | -0.0014 | 0.0462 | -0.0037 |
| | -0.0034 | 0.0414 | -0.0014 | 0.0234 | -0.0037 | 0.0512 |
| 4 | 0.0420 | -0.0042 | 0.1072 | -0.026 | | |
| | -0.0042 | 0.0444 | -0.026 | 0.1042 | | |

**Table 4.5:** Covariance matrix estimates for each estimated state of three experimental realizations of the Cy3-labelling method. The unit of each element in the matrix is $\mu m$. The SPT-$\infty$E algorithm iterated 15000 times, where 10000 iterations were burn-in. The merging threshold was set to 2-fold. Under this condition, states with diffusion coefficients that are less than 2-fold apart are merged.

| Datasets | Qdot 1 ($\mu m$) | | Qdot 2 ($\mu m$) | | Qdot 3 ($\mu m$) | |
|---|---|---|---|---|---|---|
| $\sigma_i$ | Covariance Matrix | | | | | |
| 1 | 0.7921 | -0.3102 | 0.6383 | -0.3176 | 0.9530 | -0.4232 |
| | -0.3102 | 0.7800 | -0.3176 | 0.6304 | -0.4232 | 0.9259 |
| 2 | 0.0034 | -0.0004 | 0.0024 | -0.0006 | 0.0046 | -0.0006 |
| | -0.0004 | 0.0034 | -0.0006 | 0.0025 | -0.0006 | 0.0048 |
| 3 | 0.0107 | -0.0001 | 0.0099 | -0.0012 | 0.0891 | -0.0173 |
| | -0.0001 | 0.0103 | -0.0012 | 0.01 | -0.0173 | 0.0832 |
| 4 | 0.0481 | -0.0075 | 0.112 | -0.0268 | | |
| | -0.0075 | 0.0557 | -0.0268 | 0.1224 | | |

**Table 4.6:** Covariance matrix estimations for each estimated state of three experimental realizations of the Qdot-labelling method. The unit of each element in the matrix is $\mu m$. The SPT-$\infty$E algorithm iterated 15000 times, where 10000 iterations were burn-in. The merging threshold was set to 2-fold.

$10^{-4}\mu m$ for all three experiments. This strongly suggests that the mobility of the Qdot-labelled protein is slower when compared to Cy3-labelling pro-

teins since the covariance matrix values are essentially proportional to the diffusivity. Given the size of the Qdot and as observed in past works [1, 34] Qdot-labelling hinders the mobility of the surface receptors. Therefore this result in line with expectation.

Furthermore, the results of the Qdot-labelling method are consistent among the different experiments, except for the fourth state of the first and second experiments, where the diagonal elements are approximately $0.04\mu m$ for the first experiment and 0.1 $\mu m$ for the second experiment. These inconsistencies are also present in the results for the Cy3-labelling method, which might implicate some other underlying phenomena.

An unexpected result obtained with both labelling methods and for all covariance matrix estimates was negative values in the second matrix. As we discussed in Chapter 2, the dynamical error introduces a positive correlation between the consecutive displacements, whereas the static error introduces a negative correlation between the displacements and with a value equal to the variance of the static error. Since the dynamic error correlation is proportional to $D\tau$ the absolute value is usually larger than the correlation provided by the static error when $D$ is large, yielding a final positive correlation among the displacements, and thus a positive value in the second diagonal. None of the estimated covariance matrices had a positive value, and this suggests that there may exist another component responsible for introducing a strong negative correlation among the consecutive displacements that surpasses the positive correlation of the dynamical error for any diffusivity value. It might be a component of the tracking algorithm or a component in the experimental setting. Because of that, we decided to estimate the static error as well. In this way, this unknown component is somehow incorporated into the static error and allows for the negative correlation. This is explained in more detail in the next sections.

Other important estimates from the model are the transition matrix, whose elements are the transition probabilities among states and the occupation frequency of each state. Table 4.7 shows the occupation frequency of each state. This is equivalent to the stationary probability of each state. For the Qdot-labelling method, the most occupied state was the slowest state

(first) for all three experiments. This state has covariance matrix elements around $10^{-4}$ $\mu m$, whereas, for the Cy3-labelling method, the first and second states are the most occupied ones. For the first experiment, the difference between the frequencies of these two states is less than 0.05, and the second state had covariance matrix elements around 0.008 $\mu m$. For the other two experiments, this difference was approximately 0.2, where for the second experiment, the most occupied state was the second state with covariance matrix elements approximately 0.0064 $\mu m$, and for the third experiment, the first state was the most occupied one with covariance matrix elements approximately 0.0009 $\mu m$. Therefore, the mobility of Qdot-labelled proteins tends to be slower than Cy3-labelling proteins, suggesting that the Qdot label impairs the mobility of the receptors. Moreover, in the third experiment, the tracks were slower than in the other two experiments for both labelling methods. Besides that, the fourth state in the first two experiments was rare, with approximately 0.4% of the displacements for both labelling methods, but with a higher occupation rate for Cy3. This might reflect tracking errors. Tracking errors are related to linking different particle tracks into the same trajectory. One common error is to join two positions that are very far apart in the same trajectory, implicating large displacements, generating a large diffusivity.

| **Datasets** | **Cy3 1** | **Cy3 2** | **Cy3 3** | **Qdot 1** | **Qdot 2** | **Qdot 3** |
|---|---|---|---|---|---|---|
| States | Occupation Frequency | | | | | |
| 1 | 0.4647 | 0.3520 | 0.5791 | 0.4659 | 0.5008 | 0.5839 |
| 2 | 0.4928 | 0.5350 | 0.3769 | 0.4070 | 0.3319 | 0.4114 |
| 3 | 0.0162 | 0.1082 | 0.0439 | 0.1160 | 0.1634 | 0.0044 |
| 4 | 0.0262 | 0.0047 | | 0.0111 | 0.0039 | |

**Table 4.7:** Occupation frequency of each state for each one of the experiments, estimated using SPT-$\infty$E and 15000 iterations, where the first 10000 iterations were burn-in.

One interesting result consistent among the experiments and labelling methods is that the occupation rate of the first two states (the slowest states) comprises more than 85% of the data, indicating that most receptors are in

a regime of relatively slow mobility.

Table 4.8 shows the transition matrices for each experiment. The values for state 1 for all experiments are in agreement. The diagonal elements are very similar within the same labelling methods and across different experiments. We also observe that for both Cy3 and Qdot-labels, the probability of transition to the fourth state is very rare.

| **Cy3 1** ($K = 4$) | | | | **Cy3 2** ($K = 4$) | | | | **Cy3 3** ($K = 3$) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Transition Probability Matrix | | | | | | | | | | |
| 0.7473 | 0.2044 | 0.0032 | 0.0452 | 0.6792 | 0.2578 | 0.0547 | 0.0084 | 0.647 | 0.3003 | 0.0527 |
| 0.2165 | 0.7791 | 0.0032 | 0.0012 | 0.1883 | 0.7091 | 0.1021 | 0.0005 | 0.2161 | 0.7609 | 0.0230 |
| 0.0959 | 0.0943 | 0.8095 | 0.0002 | 0.095 | 0.5914 | 0.3136 | $10^{-5}$ | 0.1789 | 0.5763 | 0.2448 |
| 0.3548 | 0.4743 | 0.0001 | 0.1708 | 0.4027 | 0.1790 | 0.1010 | 0.3173 | | | |
| **Qdot 1** ($K = 4$) | | | | **Qdot 2** ($K = 4$) | | | | **Qdot 3** ($K = 3$) | | |
| Transition Probability Matrix | | | | | | | | | | |
| 0.9128 | 0.0714 | 0.0039 | 0.0119 | 0.9147 | 0.0593 | 0.0202 | 0.0057 | 0.929 | 0.0678 | 0.0031 |
| 0.0876 | 0.8877 | 0.0174 | 0.0073 | 0.0945 | 0.8769 | 0.0282 | 0.0003 | 0.0972 | 0.8991 | 0.0037 |
| 0.0197 | 0.0711 | 0.9054 | 0.0038 | 0.0606 | 0.0628 | 0.8766 | 0 | 0.3208 | 0.4367 | 0.2425 |
| 0.243 | 0.3776 | 0.1854 | 0.194 | 0.3629 | 0.2281 | 0.1672 | 0.2418 | | | |

**Table 4.8:** Transition Matrix for each one of the experiments.

**From covariance to diffusivity**

Using the static error for each labelling method as in Chapter 2, we can recover the diffusivity of each state from Equation (4.5). The standard deviation of the static error is measured experimentally to be $\sigma_{cy3} = 0.036\mu m$ for Cy3-labels, $\sigma_{qdot} = 0.029\mu m$ for Qdot-labels. Using the covariance estimates for each state for the datasets as before, we obtain the following values for diffusivity shown in Table 4.9.

We obtain negative values for the diffusivity of some states when the relevant covariance matrix has small values on the diagonal. If the covariance matrix values are smaller than $2\sigma^2$, then the value of diffusivity is negative. This is not possible! Therefore, instead of fixing the static error, we can

| **Diffusivity**$(\mu m^2/s)$ | | |
|---|---|---|
| **Cy3 1** $(K = 4)$ | **Cy3 2** $(K = 4)$ | **Cy3 3** $(K = 3)$ |
| -0.0245 | -0.0320 | -0.0295 |
| 0.1413 | 0.0942 | 0.1586 |
| 0.9259 | 0.5027 | 1.0794 |
| 0.9754 | 2.5893 | |
| **Qdot 1** $(K = 4)$ | **Qdot 2** $(K = 4)$ | **Qdot 3** $(K = 3)$ |
| -0.0220 | -0.0258 | -0.0180 |
| 0.0425 | 0.0178 | 0.0722 |
| 0.2232 | 0.2034 | 2.1638 |
| 1.1490 | 2.7306 | |

**Table 4.9:** Diffusivity estimated using a fixed value for the static error- $0.036\mu m$ for Cy3-labelled molecules and $0.029\mu m$ for Qdot-labelled molecules- and the equations (4.5).

allow it to have a different value for each state. From equation (4.5), we have:

$$Cov(i,i) = 4D\frac{\tau}{3} + 2\sigma^2, \text{ for } i = 1,2 \tag{4.14}$$

$$Cov(i, i \pm 1) = D\frac{\tau}{3} - \sigma^2, \text{for } i = 1,2 \tag{4.15}$$

Reorganizing, and simplifying to find $\sigma$, we obtain:

$$D = \frac{3\left(Cov\left(i,i\right) + 2Cov\left(i, i \pm 1\right)\right)}{6\tau} \tag{4.16}$$

$$\sigma = \sqrt{\frac{Cov(i,i)}{2} - \frac{4D\tau}{6}}. \tag{4.17}$$

We can use equations (4.16) to estimate the diffusion coefficient for each state in each experiment and its respective static error. Table 4.10 shows these estimates.

Our estimates of diffusion coefficients and static error magnitudes are consistent across experiments. The discrepancies in estimates among the

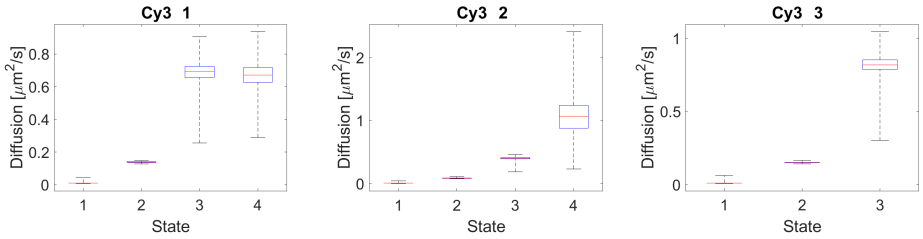
| Label | **Cy3 1** ($K = 4$) | | **Cy3 2** ($K = 4$) | | **Cy3 3** ($K = 3$) | |
|---|---|---|---|---|---|---|
| States | D $(\mu m^2/s)$ | $\sigma(\mu m)$ | D $(\mu m^2/s)$ | $\sigma(\mu m)$ | D $(\mu m^2/s)$ | $\sigma(\mu m)$ |
| 1 | 0.0091 | 0.0243 | 0.0083 | 0.0220 | 0.0099 | 0.0224 |
| 2 | 0.1139 | 0.0430 | 0.0726 | 0.0416 | 0.1221 | 0.0451 |
| 3 | 0.5545 | 0.0990 | 0.3317 | 0.0689 | 0.6403 | 0.1008 |
| 4 | 0.5594 | 0.0951 | 0.9109 | 0.1876 | | |
| Label | **Qdot 1** ($K = 4$) | | **Qdot 2** ($K = 4$) | | **Qdot 3** ($K = 3$) | |
| States | D $(\mu m^2/s)$ | $\sigma(\mu m)$ | D $(\mu m^2/s)$ | $\sigma(\mu m)$ | D $(\mu m^2/s)$ | $\sigma(\mu m)$ |
| 1 | 0.0028 | 0.01841 | 0.0001 | 0.0178 | 0.0018 | 0.0210 |
| 2 | 0.0429 | 0.0289 | 0.0198 | 0.0283 | 0.0561 | 0.0342 |
| 3 | 0.1733 | 0.0430 | 0.1238 | 0.0495 | 0.8993 | 0.1624 |
| 4 | 0.5462 | 0.1141 | 0.9637 | 0.1911 | | |

**Table 4.10:** Diffusivity and static error estimations for the three experiments, obtained using and Equations (4.16)and (4.17).

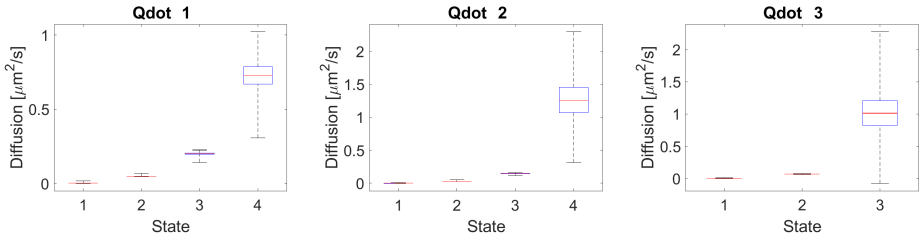
three experiments is most evident in the third and fourth states. High diffusivity states were observed for both Cy3 labelling and Qdot labelling, ranging from 0.7 to 1.1 $\mu m^2/s$ for Cy3 and 0.6 to 1.2 $\mu m^2/s$ for Qdot labelling. These values are very high compared to the usual estimates of diffusion coefficients for a surface receptor. Moreover, as we can see in Table 4.7, the states with these high values are very rare in the trajectories. Tracking errors are very likely to be the culprit for these erroneous states with very high diffusivity. Qdots are known to alternate between dark and bright levels of intensity. This is called blinking and complicates the association of positions in the tracking algorithm, which can cause the linking of different particles (tracking errors) [1, 114]. The high diffusivity values might be due to tracking errors. The introduction of large displacements in a trajectory, explained by the association of different particles, increases the diffusivity value. Moreover, the high diffusivity states are more pronounced with Qdot labelling compared to Cy3 labelling. In fact, there are two Qdot experiments where the most mobile state has D of approximately $\sim 1$ $\mu m^2/s$.

Figure 4.8 shows the posterior distribution of each estimated diffusion coefficient for the six datasets. For the highest diffusivity state of each

dataset, we also have a very large variance. The reason for that is the low number of data points from this state, as can be seen from Table 4.7. The table shows that only $\sim$ 3% to 4% of data points are in the most mobile state on average.




**(a)** Cy3



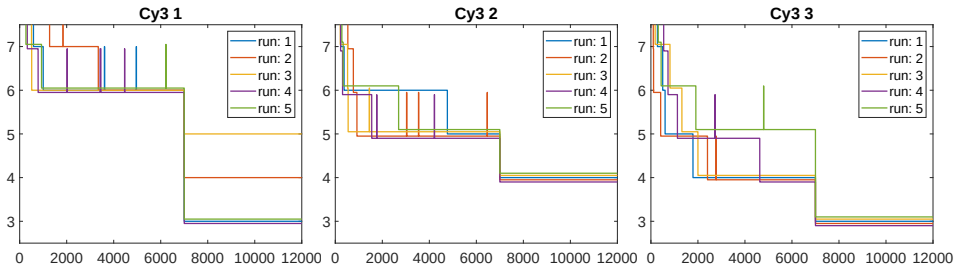

**(b)** Qdot

**Figure 4.8:** Diffusion estimates using SPT-$\infty$E from six sets of experimental trajectories, obtained by labelling IgG receptors on the surface of live B cells (A20 cell lines) either using a Cy3-labelled probe or a Qdot probe.(a) Diffusion coefficients for each state of Cy3-labelled receptors. (b) Diffusion coefficients for each state of Qdot-labelled receptors.

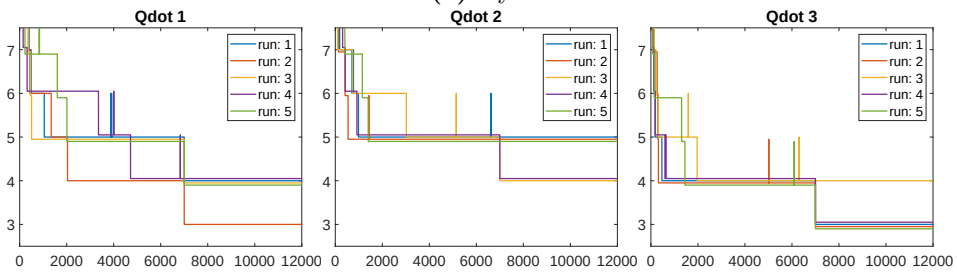
## 4.5 Robustness

To further check the performance of the algorithm on experimental datasets, we ran the full algorithm five times for each dataset. Figure 4.9 shows the number of states chain for each of the five runs. We observe that for Cy3-labelling three out of five runs converge to the same number of states for the first experiment, and on the second and third experiments all runs converge to the same number of states, four and three respectively. Qdot-experiments

have four out five converging for three states for the first and third experiments, and three out of three converging to five states. For both labelling methods, the majority of chains converge to three states for the first and third experiments chains. The results are consistent across the experiments, leading to similar conclusions.




**(a)** Cy3




**(b)** Qdot

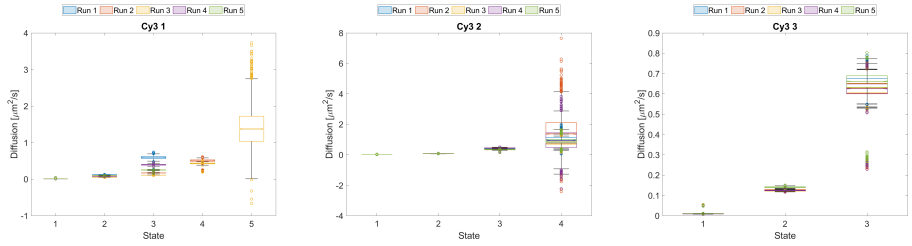
**Figure 4.9:** Five states chains of SPT-$\infty$E results from six sets of experimental trajectories obtained by labelling IgG receptors on the surface of live B cells (A20 cell lines) either using a Cy3-labelled probe or a Qdot probe. (a) Number of states for each iteration of SPT-$\infty$E on Cy3-labelled receptors. (b) Number of states for each iteration of SPT-$\infty$E on Qdot-labelled receptors.

From each of these five chains, we obtain a diffusivity posterior for each state of the chain. Figure 4.10 shows these distributions. Each box represents a diffusion distribution for each state. On each plot, we have distributions for each of the five runs and their states. The different runs are grouped by colour, and the states are indexed on the x-axis with their diffusion values on the y-axis. The distributions with large variances are the ones with the highest diffusivity, usually states 4 and 5. These are also the

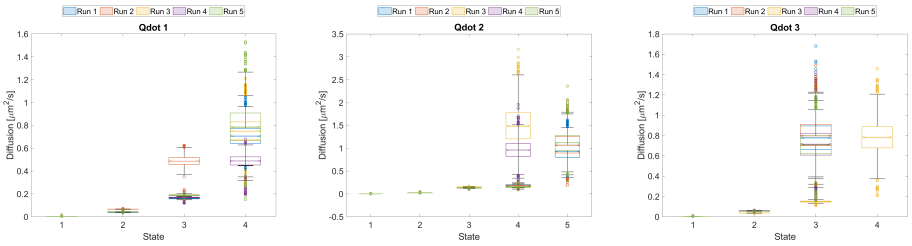
states with the smallest occupancy frequency as one can see in Figure 4.11. The smaller the occupancy frequency, the fewer data points are used to estimate the diffusion. The distributions for each state are consistent among the runs. However, we obtain some negative diffusion coefficients for the first two days of Cy3 experiments. The calculation of $D$ follows equations (4.16) and (4.17). Therefore, to have a negative $D$, the second diagonal of the covariance matrix needs to be so negative so that $\mathrm{Cov}(i,i)+2\mathrm{Cov}(i,i\pm1)<0$, given $\mathrm{Cov}(i,i)>0$ for all covariance estimations. This means that the estimated covariance matrix for the first two experiments of Cy3 has a very negative second diagonal, indicating strong anti-correlation among consecutive displacements. As we discussed before, anti-correlation is not expected if we use the localization accuracy as a fixed static error. On top of that, estimating negative diffusion further suggests that some other phenomena should be modelled in order to fully explain these data. It is interesting to note that no negative diffusion is estimated for Qdot-labelled molecules.

The occupancy frequencies of each state are consistent within runs and for two of the three of Cy3-labelling datasets and all three Qdot datasets. Even though the runs do not converge to similar occupation frequencies for the first experiment with Cy3-labelling, the results are consistent with the other two experiments. The main conclusion from Figure 4.11 is that the most occupied state for Qdot-labelled molecules is the first state, i.e. the slowest one, whereas Cy3-labelled molecules generally exist in the second state. The slowest state for Qdot-labelled molecules for all experiments was approximately $10^{-3}\mu m^2/s$, whereas Cy3-labelled molecules' slowest states was approximately $10^{-2}\mu m^2/s$, ten times faster. Again, we propose that the mobility of Qdot-labelled molecules is impaired due to the size of the Qdot.

Furthermore, we wanted to check if the mixing was good. To do this, we ran the algorithm for 50000 steps and looked at the diffusivity distribution using only the last 10000 steps. We obtain similar results as before, and the high variance for the highest diffusion coefficients (less probable states) is kept, suggesting that the algorithm is converging, properly.




**(a)** Cy3



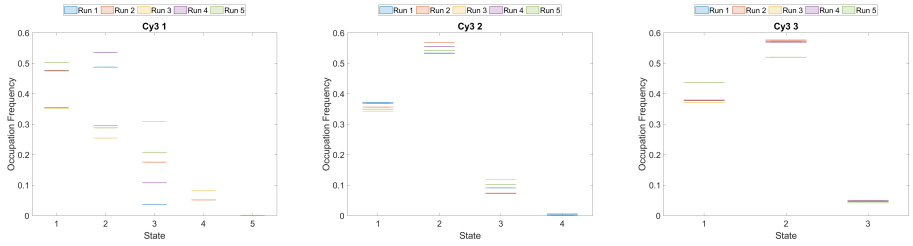

**(b)** Qdot

**Figure 4.10:** State diffusivity distribution for each dataset of experimental trajectories obtained by labelling IgG receptors on the surface of live B cells (A20 cell lines) either using a Cy3-labelled probe or a Qdot probe. Colours corresponds to different chains (runs) of the algorithm. We index the states on the x-axis, and diffusion coefficient on the y-axis. Not all runs converge to the same number of states. (a) Cy3-labelled proteins (b) Qdot-labelled proteins.

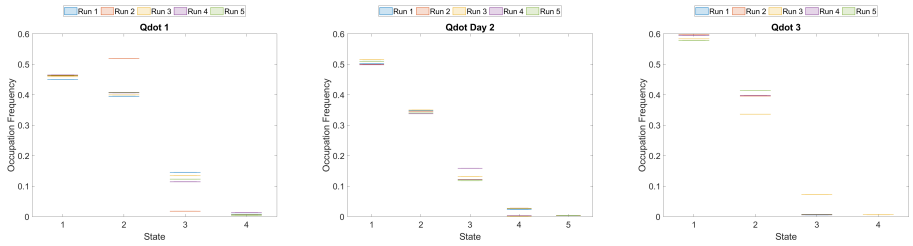
## 4.6 Summary

In this chapter, we developed SPT-$\infty$E, a Bayesian framework to estimate the number of diffusive states and their parameters, simultaneously, of single-particle tracking datasets, taking into consideration the experimental errors. We first validated the algorithm with three sets of simulated data: a 2-state dataset, a 3-state dataset and a 4-state dataset. In this framework, we introduced a merging state step after the burn-in phase to accelerate the convergence speed. This merging state step is based on a distance metric between the covariance matrix of each state.

After the validation of the algorithm, we applied the SPT-$\infty$E in the




**(a)** Cy3

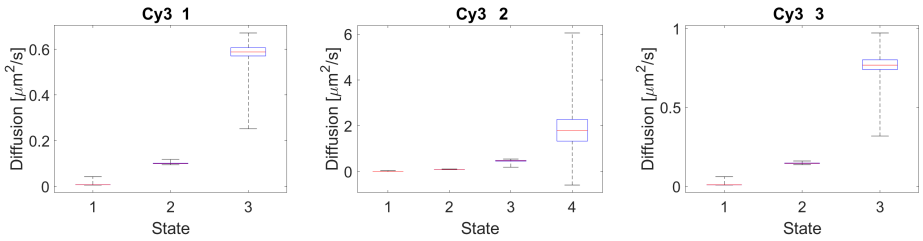



**(b)** Qdot

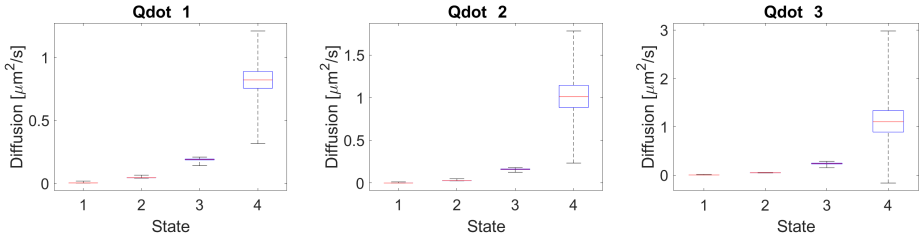
**Figure 4.11:** Boxplots of state occupation frequencies for each dataset of experimental trajectories, obtained by labelling IgG receptors on the surface of live B cells (A20 cell lines) either using a Cy3-labelled probe or a Qdot probe. Colours corresponds to different chains (runs) of the algorithm. We index the states on the x-axis, and their occupation frequency on the y-axis. Not all runs converge to the same number of states. (a) Cy3-labelled proteins (b) Qdot-labelled proteins.

same experimental datasets used in Chapter 2 and Chapter 3. From the covariance matrix estimation of each state, we encountered that fixing the standard deviation of the static error leads to negative apparent diffusion coefficients. The static error standard deviation was fixed to values measured in previous work [1]. We decided to allow the standard deviation of static error to vary among states. We solved the linear systems of equations for the covariance matrix and obtained (4.16) and (4.17).

When performing the validation of SPT-$\infty$E, we used simulated pre-localized trajectories. In practice, we did not assess the correctness of the measurement process. A way to assess this is by simulating the raw data




**(a)** Cy3




**(b)** Qdot

**Figure 4.12:** State diffusivity distribution for each experimental dataset obtained by labelling IgG receptors on the surface of live B cells (A20 cell lines) either using a Cy3-labelled probe or a Qdot probe. Colours corresponds to different chains (runs) of the algorithm. The algorithm ran for 50000 steps, but only the last 10000 iterations were used to calculate the diffusivity distribution. We index the states on the x-axis, and their diffusion coefficient on the y-axis. Not all runs converge to the same number of states. (a) Cy3-labelled proteins (b) Qdot-labelled proteins.

originated by a microscopy, and, by doing so, more realistic noise could be incorporated into the traces, such as pixelated point-spread function, and detector noises. Next, the detection and tracking algorithm performance together with SPT-$\infty$E and the measurement process could then be evaluated. This analysis could clarify the measurement process and help us to understand the apparent dependence of $\sigma$ with the diffusivity of each state as suggested by the results from experimental data.

# CHAPTER 5

## A pipeline to analyse two colour data: single-particle tracks of antigen proteins on a lipid bilayer and bulk surface receptors on cell membrane.

In this chapter, we describe a framework developed to analyze a type of experiment, where single-molecule tracking of cell-surface receptors is augmented by a second-colour labelling of a potentially interacting surface molecule. While the tracked receptors are labelled at low densities as in previous chapters, the second molecule is labelled at high density with the second colour. When the cell is imaged, the single-particle tracks appear to move over the underlying density distribution of the second molecule.

We can imagine many possible systems that could be studied under this general paradigm. Here, we will look at a system where a cell interacts, via its cell-surface receptors, with a lipid bilayer that has been augmented with ligands for the tracked receptors.

The general scheme of the experimental setting is shown on Figure 5.1. The lipid bilayer is used as a way to emulate a cell membrane enabling the study of interactions between receptors on the cell membrane with those present on the lipid bilayer [48, 65, 134]. The thickness of a lipid bilayer is

smaller than the penetration of the evanescent wave in a TIRF microscopy, making it possible to image molecules on the cell membrane, even through the bilayer.

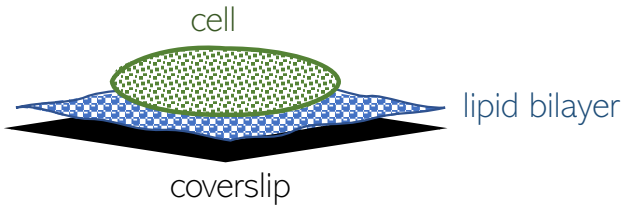



**Figure 5.1:** Scheme of experimental setup. The supported lipid bilayer emulates the membrane of a second cell. Upon adding specific cellular membrane molecules to the lipid bilayer and placing another cell on top of it, we can emulate interactions of cellular receptors between different cells.

The output of the experiment is an image stack, where each frame has two channels. One channel corresponds to the molecules on the lipid bilayer, and the other corresponds to the molecules on the cell. Here, molecules on the cell are labelled at high density, enabling the location of the cell on the image and semi-quantitative determination of molecular densities on the cell. Molecules on the lipid bilayer are labelled at low-density to allow for tracking.

The framework to analyse this type of experiment is described in the next sections. Figure 5.2 illustrates the structure of this framework, and the different pathways needed for the analysis for each image channel.

## 5.1 Biological Background

T cells play an essential role in adaptive immunity, through regulation of immune responses and direct killing of infected cells [86]. Because of that, it is of great interest to develop a deep understanding of the T cell activation process. In this work, one of the components of the system is a T cell, whose surface receptors are labelled at high density, identifying the

placement of the cell. Moreover, labelled peptide-major-histocompatibility complexes (pMHC) are added to the lipid bilayer. Certain pMHC are known to bind to T cell receptors (TCR) on a given cell, inducing receptor signalling and ultimately, cell activation - a critical step of an immune response. Signal induction is driven by molecular interactions between the T cell receptors and the pMHCs. These pMHCS are surface-bound antigens that are presented on antigen-presenting cells (APC) of the immune system. Together, pMHCs and TCRs play an essential role in the activation of T cell.

One of the major goals of cellular immunology has been to better understand which features of the TCR-pMHC interaction lead to robust cellular activation. Moreover, T cells identify small numbers of agonist pMHCS in a sea of self-pMHCs to trigger cell activation for antigen-specific immune response. In other words, T cell activation is specific and sensitive. The mechanisms governing TCR ligand discrimination remains unclear. This is a major unanswered question in immunology [15, 45, 108, 129], which has become increasingly important as strategies to therapeutically exploit cellular immunity have garnered attention, especially in the context of cancer immunotherapy [127].

Here, we describe a pipeline developed to analyse the data from a model of this system. In particular, we quantify the mobility of individual pMHC ligands on a flat lipid bilayer during interactions with TCRs. We also look at other aspects of the immediate cellular response to pMHC binding, in terms of TCR enrichment, by checking for correlations between local TCR density and the mobility of the (nearby) pMHCs.

## 5.2 High density data

In this section, we will describe the analysis path for the high-density data of the experiment. For the bulk TCRs labelled, we have an image stack of masks of the cell. However, we observe that during the whole experiment, the cell barely moves, therefore the mask is essentially immobile. The time duration of each image stack is around 10 s.

### 5.2.1 Binary Mask

The first part of the pipeline for the high-density data is to transform our TCR labelling data into a binary mask. Then, we have a way to identify whether the pMHC tracks are within the cell contact area or are outside this region. This is important because the pMHC can interact with the proteins on the cell membrane only within the cell contact area.

The image of the cell is a grey scale image, and to transform this image into a binary image, we apply some known algorithms of image processing.

First, we apply the Otsu method to define an intensity threshold of the image [92]. Then, for every pixel with an intensity larger than this threshold a value of 1 is applied and the pixel is classified as part of the cell, otherwise the pixel is on the outside of the cell, and a value of 0 is given to it. Then, we obtain a black (0) and white image (1).

The grey scale image gives a pixel intensity distribution, $\pi$. One can see an example of this image in Figure 5.3(a). To transform this image into a binary image, we need to divide this distribution into two classes: one whose intensity is large enough to be classified as cell mask and the other with lower intensity. The Otsu method finds this threshold, $t$, by minimizing the intra-class variance of the pixel intensity distribution. The intra-class variance is defined as [92]:

$$\sigma(t)_w^2 = w_0(t)\sigma_0^2(t) + w_1\sigma_1^2(t),$$

where $\sigma_0$ is the variance of the class with pixel intensity small enough to be transformed into 0 on the binary mask, and $\sigma_1$ is the class with large pixel intensities. The weights $w_0$ and $w_1$ are equal to the sum of all pixel intensities $\pi$ within each class. Since the pixel intensity distribution is normalized, $w_0 + w_1 = 1$. By minimizing the intra-class variance, we select a threshold that clearly distinguishes the classes.

After the transformation from grey scale image to binary, some pixels inside the mask are 0 even though they are inside the cell (see Figure 5.3(b)). This could be due to noise, or perhaps, there were no labelled TCRs at this location. Moreover, we also can see some pixels that are outside the cell

contact area whose intensity is classified as one. This can also be caused by noise. We assume that there are no holes in the cell-contact area. In other words, we are assuming that the cell membrane does not have any discontinuities. Therefore, even dark pixels inside the cell-contact area should be considered a cell-contact area region.

To achieve that we apply an algorithm to change to 1 all pixels inside the cell contact area. We use the *imfill* function of MatLab, a fill holes algorithm. Figure 5.3(c) shows the output image after the application of the fill holes algorithm. Here, we assume that the cell is a simply connected region, therefore if a particle is within the boarder of the cell contact area, then it is inside the cell contact area.

The next step is to smooth the border of the cell, and to delete every pixel outside the cell contact area. We use an erode algorithm to do that. This algorithm superimposes the original binary image on another image smaller than the original one. When the new image is completely contained on the original image, the pixel is retained. Otherwise, deleted [50]. For example, suppose we have a binary image A given by

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix},$$

and B defined as the smaller image $B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$. The center pixel of B is its origin. We superimpose the center of B in every pixel of the image A. If B is completely contained on A, then we keep the pixel of A, otherwise it

is set to zero. After that, the final image $C$ is:

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Finally, we dilate the image so it has a more smooth boundary. The dilation operation combines the two images. Rather than deleting the pixel when the new image is not fully contained in the original image, the dilation algorithm adds the pixel to the original image when the center of the smaller image is contained in the original image. Using $C$ and $B$, we obtain the following image after the dilation operation:

$$D = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

Figure 5.3(e) shows the final binary mask image.

### 5.2.2 Intensity Analysis

To investigate whether the organization of labelled TCR changes around tracked pMHCs, we need to estimate the density of bulk TCR near a pMHC that is interacting and possibly bound to a TCR. To decide whether a pMHC track is in a such state of interaction and bound, we applied an algorithm to classify every time point of a track as either a slow or fast state. We hypothesize that the slow states we identify are the possible bound (interacting) state. This is discussed and explained in the next sections.

Once we have the pMHC mobility states, we focus on the slow states and calculate the TCR density in the vicinity. First, we determine a number $n$ of pMHCs in the slow state within the cell contact area. We calculate the ratio

of the average intensity of the third neighbour pixels to the average intensity of the pMHC pixel and its first and second neighbours (Figure 5.4). If this ratio is larger than one, it indicates that TCRs are locally depleted close to the pMHC, and if it is smaller than one, it indicates enrichment of TCR in the neighbourhood of the pMHC. Since we calculate this ratio for every image of the stack, we obtain a distribution. We also generate a background intensity ratio distribution as a baseline control. For the calculation of the control distribution, we randomly choose pixels of each frame and calculated the same ratio of intensity. As a result, we obtained a control distribution that provides a base for comparison with the experimental results.

Finally, for the bulk TCR, we test the null hypothesis of both distributions having the same mean and the alternative hypothesis that the mean of the bulk TCR is larger than the control. We used the Kolmogorov-Smirnov test (KS test) to test whether the control sample and the slow-state intensity sample have the same distribution. These tests allow us to demonstrate if TCR aggregation around potentially bound pMHC is occurring.

## 5.3 Low density data

In this section, we explain the step-by-step of the analysis of the low density data, the single-particle trajectories.

### 5.3.1 Detection and Tracking of pMHC trajectories

As in previous chapters, we begin by transforming the raw data into trajectories. First, we run an algorithm to detect the position of each particle. After, we need to link these particles into trajectories. Particles were localized and tracked by Icy bioimaging analysis software. Particle detection was done using undecimated wavelet transform method, whose settings were: detection of bright spots over dark background with scale 2, allowing the detection of spots about 4 to 7 pixels of diameter and threshold 70. Particle tracking was done using the multiple hypothesis tracking method. Diffusion and directed movements were considered in the tracking as well. The out-

put files consisted of the frame number and x-y positions of each particle as explained on the introduction.

Figure 5.5 shows an image with the trajectories generated by the detection and tracking algorithms. Besides that, it also shows the binary mask and how we use it sto classify the tracks. On the right image, we have red trajectories from tracks that are outside the cell contact area during the whole capture of the image. The green tracks are the ones that cross the boundary of the cell overlapping region, and blue tracks remain entirely inside the cell region.

### 5.3.2 Localization precision of pMHC positions

To measure the localization accuracy of the experiment, immobile particles were placed on the coverslip. Then, the same imaging detection and tracking techniques were applied. We estimated the localization precision from the displacement distribution of the immobilized tracks. The standard deviation of this distribution is the localization precision of the measurement and was found to be 53 nm.

### 5.3.3 Single-State Analysis of pMHC trajectories

To assess the mobility of the pMHC tracks, we first applied a single state diffusion analysis. We assumed the tracks were under a two dimensional Brownian diffusion. A maximum likelihood estimator [6] was used to estimate the diffusion coefficient ($D$). This estimator takes in consideration the positional errors and blurring during image acquisition. The localization precision (Chapter 1) gives the positional error which is used in the method. However, we also applied this method to estimate both diffusion coefficients and the localization precision. The results for the latter were similar to those acquired from immobile particles.

### 5.3.4 Constrained 2-states Analysis of pMHC trajectories

The constrained 2-states model is based on the model described in Das et al [24] where we have transitions between two states: *slow* and *fast*. However,

since the output data contains particles moving outside and within the cell contact area, we further improve the model to constrain the particles that are outside to not transit to slow states. The reason for this is: pMHCs outside the cell contact area cannot bind to any other proteins, and therefore their diffusive process should not change its regime.

We have two processes for the constrained Model:

- The particles **outside** the cell area undergo simple Brownian motion: state 1.
- The particles **inside** the cell area follow a Markov process with 2 diffusive states: a fast state 2 and a slow state 3.

Figure 5.6 illustrates the constrained model. Figure 5.6(a) is an illustration of the model. Each "X" is a track on the frame. The track can either be in a slow or in a fast state, when within the cell contact area. Otherwise, the track can only be in the fast state. Figure 5.6(b) is a sketch of the Markov process that occurs within the cell contact area, the diffusive process on the outside of the contact region, and their relations.

Following the same procedure as in Das et al [24], the parameter estimation is done by maximizing the likelihood function via a MCMC algorithm. The likelihood for the constrained model is given by:

$$\mathcal{L}(\theta|\mathbf{O}) \propto \sum_{\mathbf{q_1},\cdots,\mathbf{q_{3^N}}} P(\mathbf{O}|\mathbf{q_i},\theta)P(\mathbf{q_i}|\theta), \tag{5.1}$$

where $\{q_{i=1}^N\}$ is a sequence of $N$ binary state variable and the likelihood is proportional to the sum of probability of all the $2^N$ possible sequences of $\{q_i\}_{i=1}^N$. Since this consists of an extensive computation, we applied the forward-backward algorithm [24, 100].

### 5.3.5 Forward Algorithm for the Constrained Model Likelihood

The forward-backward algorithm uses the forward probability, $\alpha_j(i)$, to build the log-likelihood step by step. The forward probability $\alpha_j(i)$ is

the probability of observing the partial sequence of displacements, $\mathbf{O} = \Delta_1, \Delta_2, \cdots \Delta_j$, where displacement $\Delta_j$ is in state $i$, given the parameters $\Theta$. The likelihood for a sequence of displacements $\Delta_0, \Delta_1, \cdots, \Delta_n$ of the constrained 2-states model is given by

$$\mathcal{L}(\theta|\mathbf{O}) = \sum_{k=1}^{3} \alpha_N(k).$$

The forward probability is given by

$$\begin{aligned}
\alpha_j(i) &= \left[\sum_{k=2}^{3} \alpha_{j-1}(k) p_{ks_j}\right] P\left[\Delta_j | s_j = i, \Theta\right] (1 - \delta_{i1}) \left(1 - \delta_{s_{(j-1)}1}\right) \\
&+ \left[\sum_{k=1}^{2} \alpha_{j-1}(k)\right] P\left[\Delta_j | s_j = i, \Theta\right] \delta_{i1} \\
&+ \alpha_{j-1}(1) P\left[\Delta_j | s_j = 2, \Theta\right] (1 - \delta_{i1}) \, \delta_{s_{(j-1)}1},
\end{aligned}$$

where $P\left[\Delta_j | s_j = m, \Theta\right] = e^{-\frac{\Delta_j^2}{2D\tau}}$ with $D$ equal to the diffusion coefficient of the slow state for $m = 3$ and equal to the diffusion coefficient of the fast state for $m = 1, 2$. $\delta_{ij}$ is the Kronecker delta, which is 1 when $i = j$ and zero otherwise. We are using the Kronecker delta to consider all the possible boundary-crossing events and their respective likelihood: the particle is outside in the current step, and the particle is outside in the previous step and inside in the current one.

## 5.4 Results

Here, we apply our analytical pipeline to unpublished experiments. In these experiments, supported planar bilayers were constructed containing low densities of labelled pMHC. T cells bearing either labelled CD45 or labelled TCR were allowed to settle on the bilayer and cell-bilayer junctions were imaged using two-colour total internal reflection fluorescence microscopy (TIRF). In this work, we use the pipeline to study the interaction of TCR with several peptide ligands, in the context of T cell interactions

with a supported lipid bilayer.

### 5.4.1 Single-state Analysis

To detect any changes in pMHC mobility induced by the presence of the T cell and whether these changes are common within different pMHC ligands, we first apply the single-state analysis taking into account experimental errors [6] for the pMHC tracks.

We begin by classifying the pMHC tracks into three groups - those entirely outside the cellular mask area, those entirely within the mask, and those that entered or left the mask during the experiment. We estimate the effective diffusion constants for the first two groups only, neglecting those that crossed the mask boundary.

In Figure 5.7, we present detailed results for one set of experiments performed with cells bearing the AND TCR, interacting with four different pMHC ligands: (i) the "wild-type" agonist ligand moth cytochrome c (MCC); (ii) the low-affinity peptide variant T102S; (iii) another peptide variant K99A with very low affinity; (iv) and $\beta$2m which is present on the same MHC, but is an irrelevant peptide that TCR do not bind. In the context of ligands, an agonist is a ligand that binds to a receptors and produce a biological response leading to cell activation in this particular case the binding of pMHC to TCRs.

We immediately observed that the cell-independent (outside-mask) diffusivities of all four peptides were similar, but not identical, probably reflecting variation due to the construction of supported bilayers. Moreover, diffusivities outside of the T cell area were mostly larger than those inside. This suggests that the presence of the T cell induces a decrease in pMHC mobility. This could be both due to nonspecific interactions with surface molecules on the T cell as well as binding to TCRs.

Additionally, the presence of the cell substantially alters particle mobility in a pMHC-dependent manner. The MCC pMHC shows the largest change in diffusivity. This pMHC is known to bind to the TCR present on the T cells. By comparison, the $\beta$2m pMHC shows a small change in median diffusivity.

This is known to be a weakly-interacting pMHC, so the change in mobility possibly reflects nonspecific interactions between the MHC molecule and CD4 on the T cell membrance, and/or steric hindrance due to the close apposition of cell and bilayer [1]. We also found that the weak agonist pMHC T102S consistently exhibited intermediate retardation, while motion of the K99A molecule was minimally retarded. We can interpret these results as showing that pMHCs bind to TCR on the T cell surface, reducing their mobility in a manner that reflects their known potency as TCR agonists. Table 5.1 summarizes the single-state diffusivity results for AND T. The median diffusivity for the pMHC inside and outside the cell contact area is reported.

### 5.4.2 5cc7 T cells

To examine a different TCR, we also performed experiments using labelled 5cc7 T cells, with three pMHC variants in the bilayer: a super-agonist K5, wild type MCC and a weak agonist T102S. Results for these systems, from experiments performed on two different days, are shown in Figure 5.8. In this case, the mobility of the MCC peptide was still substantially reduced by the presence of the T cell, but the effect is not as dramatic. This is consistent with our understanding that the AND TCR has higher affinity for MCC pMHC than the 5cc7 TCR. Intriguingly, the pMHC T102S does not exhibit intermediate retardation when in contact with 5cc7 cells. This distinction between AND and 5cc7 cells is unexpected, and the reason for the difference on diffusivity is unclear.

Super-agonist K5 exhibits some retardation of diffusivity by the presence of the T cell. However, this retardation is not as pronounced as the wild-type MCC. The pMHC K5-5CC7 TCR bond is more stable and shorter-lived than the pMHC WT-5cc7 TCR bond [108]. Thus, it is expected that the impact of the presence of the T cell on K5 diffusivity should be larger than WT MCC pMHC. It is therefore unclear why the K5 diffusivity is not affected as much.

Table 5.1 summarizes the results for all experiments with AND T cells

| TCR | pMHC | $D_{out}$ ($\mu$m$^2$/s) | $D_{in}$ ($\mu$m$^2$/s) | Fold change to $D_{out}$ | # of experiments |
|---|---|---|---|---|---|
| AND | MCC | 0.65 | 0.03 | 36 | 6 |
| | T102S | 0.69 | 0.18 | 4.5 | 5 |
| | K99A | 0.89 | 0.52 | 1.6 | 3 |
| | $\beta$2m | 1.05 | 0.29 | 3.8 | 2 |
| 5cc7 | MCC | 0.44 | 0.10 | 6.2 | 2 |
| | T102S | 0.64 | 0.35 | 1.8 | 2 |
| | K5 | 0.47 | 0.30 | 3.3 | 2 |

**Table 5.1:** Diffusion coefficients given here are averages of medians from the number of experiments indicated. Representative fold change reported here is the mean of the fold-change of median, over the number of experiments indicated. All experiments were performed with a TCR label.

and 5cc7 T cells. We show the median diffusivity for the pMHC inside and outside the cell contact area, the fold change between these diffusivities and the number of experiments performed.

The cell-independent diffusivities among 5cc7 T cells are significantly lower than AND T cells, and this reflects the inter-experimental variability. We observe that the highest fold change is for WT MCC pMHC tracks for AND TCRs, where the inside diffusivity is 36 times smaller than the outside diffusivity. A curious result is the super agonist K5, whose fold change is only 3.3, similar to the self-peptide $\beta$2m. It is worth to notice that both $\beta$2m and K5 were averaged over only 2 experiments. More experiments might provide a higher fold change for K5 compared to $\beta$2m.

### 5.4.3 Experiments with addition of CD80

Furthermore, we also analyzed experiments where unlabelled CD80 molecules were added to the lipid bilayer. Supported bilayers bearing ICAM-1, CD80 and labelled pMHC were generated. We tracked labelled pMHC in the presence of AND T cells. Again, the TCR was labelled in a second colour.

CD80 is a costimulatory molecule that is present on the surface of APCs. It binds to CD28 presented on the T cell surface, leading to downstream signals that play an important role in cytoskeletal alterations and the early spatial organization of cell signaling [33]. We were interested to see if the presence of CD80 altered the mobility of pMHC in the supported bilayer.

To do that, we applied the single state diffusion analysis to obtain the effective diffusivity distribution for each experiment. Figure 5.9 shows the cumulative probability distribution of diffusivity, across experiment settings: (i) WT MCC pMHC (solid) versus WT MCC pMHC with CD80 (dashed),(ii) K99A MCC pMHC (solid) versus K99A MCC pMHC with CD80 (dashed), and (iii) T102S MCC pMHC (solid) versus T102S MCC pMHC with CD80 (dashed). No significant difference between samples with and without CD80 was observed.

### 5.4.4 Experiments with addition of unlabelled pMHCs

We were also interested in seeing if the addition of unlabelled pMHC into the bilayer would lead to changes in labelled pMHC mobility. Individual pMHC might not be able to reliably induce strong TCR signals to drive T cell cytoskeletal alterations and/or TCR clustering on the cell surface. By providing additional pMHC we hoped to drive signaling and detect changes in pMHC mobility. In separate experiments, we generated supported bilayer systems bearing ICAM-1, labelled MCC pMHC, and variable additional concentrations of unlabelled MCC pMHC to the bilayer (units of unlabelled pMHC over $\mu m^2$). We then performed single particle tracking of the labelled pMHC tracks in the presence of 5cc7 T cells with labelled TCR providing the cell contact area mask information. Results of the single state diffusivity analysis are shown in Figure 5.10 for six experimental settings: (i) WT + 0 unlabelled WT versus WT + 10 unlabelled WT over $\mu m^2$, (ii) WT + 0 unlabelled WT versus WT + 100 unlabelled WT over $\mu m^2$, and (iii)WT + 0 unlabelled WT versus WT + 1000 unlabelled WT over $\mu m^2$. These numbers indicate the concentration of unlabelled peptides. Perhaps surprisingly no significant difference in the mobility between experiments with and without

additional unlabelled WT MCC pMHC was observed.

### 5.4.5 Application of constrained hidden Markov model to experimental data

The next step is to investigate possible transitions between different modes of diffusion for the individually labelled pMHC. As discussed in previous chapters, it is possible to apply an algorithm that classifies the data in two different groups of diffusivity: fast, and slow. This method seems appropriate for these experiments because of the nature of events in the system. Here, pMHCs transit in or out of the cell contact area. Inside the cell contact area, the pMHC could bind to TCRs. Applying the constrained two-state model allows us to assess the changes in pMHC mobility that may corresponds to these binding or unbinding events.

Figure 5.11 presents results for the same set of experiments (AND T cells and 5cc7 T cells) used for Figures 5.7 and 5.8. As in the previous section, we found substantial variability in the background (fast) diffusivity of pMHC in the lipid bilayer, making comparison of absolute numbers from one experiment to another challenging. However, the slow state diffusivity and transition rates present a qualitatively reasonable picture across experiments.

WT pMHCs exhibits the slowest slow diffusivity on AND T cells, around 1.5 times smaller than K99A, T102S and $\beta$2m peptides. This probably reflects the formation of bonds among WT MCC pMHC and AND TCRs. The other peptide variants and the self-peptide $\beta$2m have similar slow diffusivity. However, besides the diffusivity, the rate of binding and unbinding events gives information about the bonds as well.

We observe that WT MCC pMHC had one of lowest off rates among the variants, suggesting that once the bond forms, it is slower to break compared to other pMHC-TCR bonds studied here. Moreover, WT MCC pMHC has the largest on rate among all peptides, suggesting that it can rapidly find and bind to TCR within the contact area. In summary, WT MCC pMHC exhibits apparent bonds that form more rapidly and are more stable than the

mutant pMHCs. In Figure 5.11, the points representing K99A and T102S are below the line $y = x$ and, therefore, they tend to form a bond more than break it. However, this tendency is not as strong as WT MCC pMHC. For the self-peptide $\beta$2m, the bond is easier to form in comparison with K99A and T102S, but it is also easier to break.

For 5cc7 TCRs the slow diffusivity is similar for all peptides. The WT MCC pMHC has the largest on rate and one of the smallest off rate, similarly to AND TCRs experiments. The super agonist K5 exhibits the highest off rate and the second largest on rate, meaning the bond is not as stable as the WT MCC pMHC bonds and T102S bonds. Previous works have found that K5 and WT MCC pMHC have shorter bonds and have higher 3D in vitro binding affinities with TCRs compared to T102S [108]. Moreover, it was previously found that K5 pMHC has the shortest bond distance with TCRs [21, 54, 108]. The bond distance is defined as the distance between the center of each molecule. The shorter the distance, the more strength needed to break the bond, and the higher the energy of the bond. Then, more energy is needed to form a short bond, and to break it. Here, we observe that K5 pMHC mobility is not as affected by the presence of the T cell as the WT MCC pMHC (see Table 5.1). Moreover, the on and off rate suggest a less stable bond compared to WT. Therefore, our results for K5 are not in complete agreement with past works [21, 54, 108].

### 5.4.6 Frequency of states

Finally, we calculated the relative times that the pMHC spent in the slow state and the fast state. Figure 5.12 shows the percentage of states for each experiment. We observe that WT MCC pMHC mostly occupies the slow states with 74% occupancy for the AND TCRs and 56% for the 5cc7 TCRs. K99A and T102S have similar values 55% and 61% occupancy for the slow states and the self-peptide $\beta$2m has less than half occupancy of slow states. Both K5 and T102S presents the majority of tracks on the fast state.

### 5.4.7 Intensity Analysis

TCRs form clusters on the cell surface following stimulation by pMHC [14, 133]. Conversely, CD45 is known to be excluded from the region around TCR-pMHC bonds [20, 108, 118, 134]. We were interested to learn if TCR enrichment or CD45 exclusion could be detected in the vicinity of labelled pMHC interacting with the T cell. Here, we developed a method to assess whether there is any TCR enrichment or CD45 exclusion in the vicinity of possible TCR-pMHC bonds. As explained previously, the constrained two-state Markov model infers which state the pMHC is in at each time point. Since the slow states represent possible binding events, we decided to explore the intensity of the region around slow-state pMHC tracks.

We selected slow-state pMHC tracks from the data, and calculated the average intensity of the surface marker channel over a 25-pixel square centred on the pixel containg the focal pMHC. This averaged pMHC-proximal intensity was then normalized by dividing by the average intensity of all pixels immediately adjoining the 25-pixel square (see Figure 5.4). This numerical ratio was then compared to a control ratio obtained by treating randomly chosen pixels within the same experiment in an identical manner, allowing us to assess statistical significance. We calculated the average intensity only at tracks that have at least $n$ consecutive frames in the slow state. The reason is that we assumed that the time scale from a pMHC-TCR bond to leading TCR enrichment or CD45 exclusion is larger than the time interval between frames. Hence, we applied this only to pMHC-TCR interactions that last for at least $n$ frames (assuming that slow states represent potential interactions). We set $n$ to 50, 100 and 200 for TCR experiments, and 50 and 400 for CD45 experiments. These number were based on data, as we increase $n$, fewer tracks are available. In addition, we expect that the time scale for CD45 exclusion to be larger than TCR enrichment, and that is why we are reporting 400 frames. After calculating the intensity distribution for each track, we performed a KS-test between this distribution and the control distribution.

This analysis was performed using the AND TCR with WT, T102S, K99

and $\beta$2m pMHC. We found a small (2%-5%) but statistically significant increase ($p < 0.0016$) in TCR intensity around WT pMHC in the slow state. Using the 5cc7 TCR experiments with K5, WT and T102S experiments, we also found evidence for a small (2%-6%) TCR enrichment, and statistically significant enrichment of TCR in the vicinity of WT pMHC in the slow state ($p < 10^{-3}$). In parallel experiments, the CD45 intensity was found to exhibit a small (2%-3%) decrease around WT pMHC, and this was also significant ($p < 10^{-3}$). Results for other pMHC were not significant. Tables 5.3 and 5.5 summarize these results for TCR mask experiments and CD45 mask experiments, respectively. In the parameters row, R stands for rejection of the null hypothesis.

Tables 5.2 and 5.4 reports the results of the two-state constrained model for all experiments using both TCR masks and CD45 masks.

The results for the CD45 mask experiments are similar to the TCR masks experiments. We obtained a larger frequency of slow states for WT MCC pMHCs compared to T102S pMHCs and K99A pMHCs. Although there is difference in the results among different days of experiments, but similar pMHCs, probably reflecting the variation due to the construction of supported lipid bilayers.

For the CD45 masks, experiments with addition of unlabelled peptides were performed on WT MCC pMHC, and T102S pMHCs. For WT pMHCs, the transition probabilities ratio increased with the addition of WT pMHC. No significant difference was observed for T102S.

## 5.5 Summary

In this chapter, we presented a novel pipeline to analyse two-colour TIRF microscopy data to study the dynamics of pMHCs in a supported lipid bilayer, and their correlation with the formation of pMHC-TCR bonds. The data comprises single-particle tracking of pMHCs on a supported bilayer labelled at low-density, and high density labelling of proteins on the cellular membrane (TCR or CD45) of the cell placed on top of the lipid bilayer.

The pipeline has two pathways: the analysis of the high density channel,

| T Cell | pMHCs | $D_{fast}$ $(\mu m^2/s)$ | $D_{slow}$ $(\mu m^2/s)$ | $p_{fs}$ | $p_{sf}$ | $\frac{p_{fs}}{p_{sf}}$ |
|---|---|---|---|---|---|---|
| AND | **$\beta$2m** | 0.7167 | 0.0417 | 0.0525 | 0.0693 | 0.7580 |
| | **K99A** | 0.8117 | 0.0343 | 0.0311 | 0.0455 | 0.6840 |
| | **T102S** | 0.6897 | 0.0373 | 0.0332 | 0.0329 | 1.009 |
| | **WT** | 0.6893 | 0.0330 | 0.1062 | 0.0372 | 2.855 |
| | **K99A** | 0.5520 | 0.0408 | 0.0323 | 0.0509 | 0.6350 |
| | **K99A CD80** | 0.5711 | 0.0372 | 0.0253 | 0.0394 | 0.6420 |
| | **$\beta$2m** | 0.4309 | 0.0340 | 0.0202 | 0.0541 | 0.3730 |
| | **K99A** | 0.4637 | 0.0408 | 0.0212 | 0.0624 | 0.3400 |
| | **WT** | 0.4777 | 0.0320 | 0.0436 | 0.0230 | 1.8960 |
| 5cc7 | **K5** | 0.5105 | 0.0345 | 0.0206 | 0.0595 | 0.3460 |
| | **T102S** | 0.4068 | 0.0475 | 0.0222 | 0.0328 | 0.6770 |
| | **WT 0** | 0.4210 | 0.0310 | 0.0139 | 0.0216 | 0.6440 |
| | **WT 10** | 0.4734 | 0.0236 | 0.0289 | 0.0328 | 0.8810 |
| | **WT 100** | 0.4504 | 0.0267 | 0.0375 | 0.0514 | 0.7300 |
| | **WT 1000** | 0.4425 | 0.0323 | 0.0287 | 0.0428 | 0.6710 |

**Table 5.2:** Results of the constrained two-state model on experiments performed done with TCR mask using AND T Cells and 5CC7 T Cells. The numbers 0, 10, 100, and 1000 indicate the addition of unlabelled peptides and its concentration over $\mu m^2$.

and the analysis of the low density channel. The high density channel analysis is further divided into two pathways- (i) a binary mask of the cell to classify pMHCs that are inside the cell-contact are or outside the cell-contact area, and (ii) an intensity analysis that quantifies the average intensity ratio of pixels of selected pMHCs. The low density analysis is divided into three pathways: (i) a single state diffusion analysis comparing pMHC tracks entirely outside the cell-contact area with pMHC tracks entirely inside the cell-contact area, (ii) a two-state constrained model that allows for tracks to transition between a slow and fast state when inside the cell-contact area

| T cell | pMHC | R: 50 | mean | p-value | R: 200 | mean | p-value |
|---|---|---|---|---|---|---|---|
| AND | $\beta$**2m** | *no* | 1.0187 | 0.9991 | — | — | — |
| | **K99A** | *no* | 0.9970 | 1 | no | 1.0004 | 0.9880 |
| | **T102S** | *no* | 0.9990 | 0.9999 | no | 0.9915 | 1 |
| | **WT** | *yes* | 1.0308 | $10^{-178}$ | yes | 1.0306 | $10^{-167}$ |
| | **K99A** | *yes* | 1.0072 | 0.0012 | yes | 1.0165 | $10^{-31}$ |
| | **K99A CD80** | *no* | 1.0028 | 0.6535 | yes | 1.0038 | 0.0340 |
| | $\beta$**2m** | *no* | 0.9922 | 1 | *no* | 0.9954 | 0.9999 |
| | **K99A** | *yes* | 1.0125 | $10^{-7}$ | *yes* | 1.0179 | $10^{-27}$ |
| | **WT** | *yes* | 1.0106 | $10^{-5}$ | *yes* | 1.0084 | 0.0016 |
| 5cc7 | **K5** | *no* | 1.0056 | 0.235 | *no* | 1.0043 | 0.431 |
| | **T102S** | *yes* | 1.0127 | $6.8 \times 10^{-4}$ | *yes* | 1.0283 | $10^{-8}$ |
| | **WT 0** | *yes* | 1.0215 | $10^{-9}$ | *yes* | 1.0120 | $4 \times 10^{-4}$ |
| | **WT 10** | — – | — | — | — | — | — |
| | **WT 100** | — | — | — | — | — | — |
| | **WT 1000** | — | — | — | — | — | — |

**Table 5.3:** Results of the intensity analysis and the KS-test on experiments performed done with TCR mask using AND T Cells and 5CC7 T Cells. The numbers 0, 10, 100, and 1000 indicate the addition of unlabelled peptides and its concentration over $\mu m^2$.

(a slow state that represents a potential pMHC-TCR bond formation), and (iii) calculation of the frequency of slow states.

The main conclusion for the low-density analysis is the decrease in the mobility of WT MCC pMHCs due to the presence of the T cell. This is probably caused by both non-specific interactions among pMHCs with cell surface molecules and the formation of pMHC-TCR bonds. In addition, the presence of the cell alters particle mobility in a pMHC-dependent manner, where WT MCC pMHC shows the largest decrease on diffusivity, and

| T Cell | pMHCs | $D_{fast}$ $(\mu^2/s)$ | $D_{slow}$ $(\mu^2/s)$ | $p_{fs}$ | $p_{sf}$ | $\frac{p_{fs}}{p_{sf}}$ |
|---|---|---|---|---|---|---|
| | **WT 0** | 0.3873 | 0.0319 | 0.0193 | 0.0331 | 0.5831 |
| AND | **WT 100** | 0.4336 | 0.0313 | 0.0517 | 0.0403 | 1.2829 |
| | **T102S 0** | 0.4368 | 0.0409 | 0.0303 | 0.0454 | 0.6674 |
| | **T102S 10** | 0.4654 | 0.0391 | 0.0320 | 0.0439 | 0.7289 |
| | **T102S 100** | 0.4339 | 0.0420 | 0.0300 | 0.0511 | 0.5871 |
| | **T102S 1000** | 0.3582 | 0.0399 | 0.0297 | 0.0456 | 0.6513 |
| | **WT** | 0.2854 | 0.0311 | 0.0783 | 0.0250 | 3.132 |
| | **T102S** | 0.2673 | 0.0370 | 0.0346 | 0.0259 | 1.336 |
| | **WT** | 0.2788 | 0.0303 | 0.0736 | 0.0303 | 2.429 |
| | **K99A** | 0.4368 | 0.0452 | 0.0341 | 0.0854 | 0.3990 |
| | **T102S** | 0.4071 | 0.0386 | 0.0466 | 0.0592 | 0.7870 |
| | **K99A** | 0.4294 | 0.0758 | 0.0134 | 0.0348 | 0.385 |
| | **K99A 200 CD80** | 0.4309 | 0.0886 | 0.0182 | 0.0344 | 0.529 |
| | **T102S** | 0.4202 | 0.0697 | 0.0192 | 0.0375 | 0.512 |
| | **T102S 200 CD80** | 0.4313 | 0.051 | 0.0229 | 0.0463 | 0.495 |
| | **WT** | 0.3101 | 0.0521 | 0.0311 | 0.037 | 0.841 |
| 5cc7 | **WT** | 0.7295 | 0.0313 | 0.0953 | 0.0416 | 2.291 |

**Table 5.4:** Results of the constrained two-state model on experiments performed done with CD45 mask using AND T Cells and 5CC7 T Cells. The number 200 indicates the addition of unlabelled CD80 and its concentration over $\mu m^2$.

| T cell | pMHCs | **R: 50** | **mean** | **p-value** | **R: 400** | **mean** | **p-value** |
|---|---|---|---|---|---|---|---|
| | **WT 0** | *yes* | 0.9938 | $10^{-32}$ | *yes* | 0.9892 | $10^{-32}$ |
| AND | **WT 100** | *yes* | 1.0003 | $10^{-17}$ | *yes* | 1.0050 | 0.0057 |
| | **T102S 0** | *yes* | 1.0049 | 0.0038 | *yes* | **0.9952** | $10^{-10}$ |
| | **T102S 10** | *no* | 1.0089 | 0.9831 | *yes* | 0.9977 | $10^{-12}$ |
| | **T102S 100** | *no* | 1.0077 | 0.8819 | *no* | 1.0118 | 0.9999 |
| | **T102S 1000** | *no* | 1.0106 | 0.9867 | *no* | 1.0021 | 0.0594 |
| | **WT** | *yes* | 0.9966 | $1 \times 10^{-4}$ | *no* | 1.0090 | 1 |
| | **T102S** | *no* | 0.9998 | 0.5217 | *no* | 1.0527 | 1 |
| | **WT** | *no* | 1.0169 | 1 | *no* | 1.0255 | 1 |
| | **K99A** | *no* | 1.0105 | 1 | *no* | 1.0208 | 1 |
| | **T102S** | *no* | 0.9984 | 0.6447 | *no* | 0.9974 | 0.7940 |
| | **K99A** | *no* | 1.0189 | 1 | *no* | 1.0070 | 0.4063 |
| | **K99A 200 CD80** | *no* | 1.0137 | 1 | *no* | 1.0055 | 0.3986 |
| | **T102S** | *no* | 1.0167 | 1 | *yes* | 0.9896 | $1 \times 10^{-14}$ |
| | **T102S 200 CD80** | *no* | 1.0084 | 0.9964 | *no* | 1.0086 | 0.9249 |
| | **WT** | *yes* | 0.9991 | $6 \times 10^{-5}$ | *yes* | 0.9858 | $1 \times 10^{-17}$ |
| 5cc7 | **WT** | *yes* | 0.0855 | $10^{-13}$ | *yes* | 0.0850 | $10^{-14}$ |

**Table 5.5:** Results of the intensity analysis and the KS-test on experiments performed done with CD45 mask using AND T Cells and 5CC7 T Cells. The number 200 indicates the addition of unlabelled CD80 and its concentration over $\mu m^2$.

K99A, which has the lowest affinity for the AND TCR, the smallest decrease for AND T cells. For 5cc7 T cells, WT MCC shows the largest decrease, and T102S the smallest decrease. In addition, the two-state constrained algorithm reports the transition probabilities between fast and slow states, which can be used to calculate the $k_{on}$ and $k_{off}$ rates for the pMHC-TCR bond formation reaction. We observed that WT MCC pMHC exhibit apparent bonds that form more rapidly and are more stable. However, some results are not in complete agreement with previous reports, particularly with regard to K5 pMHC on 5cc7 cells [21, 54, 108].

For the high-density analysis, we detected a small (2%-6%) but statistically significant increase in TCR intensity around WT pMHCs in the slow state for both AND and 5cc7 cells. For the CD45 mask experiments, we found a small (2%-3%) but significant decrease of CD45 intensity. These results suggests TCR enrichment in the vicinity of a bond (clustering), and exclusion of large molecules from the vicinity of a pMHC-TCR bond, in agreement with previous works [14, 20, 108, 118, 134]. These small increases (or decreases) might be due to the TIRF microscopy not having enough resolution to capture the spatiotemporal scale of this phenomenon.

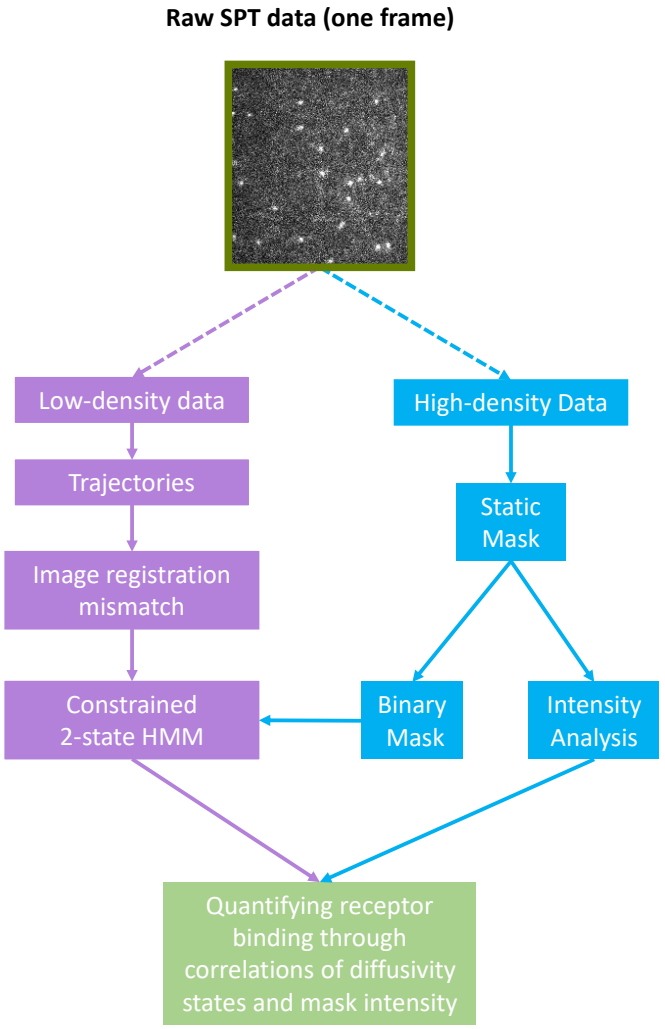



**Figure 5.2:** Pipeline scheme showing the steps for each data. Purple boxes shows the analysis on the low-density data that is from getting the trajectories to the diffusion analysis using the constrained 2-state HMM. Blue boxes shows the pathway to analyse the high-density data.

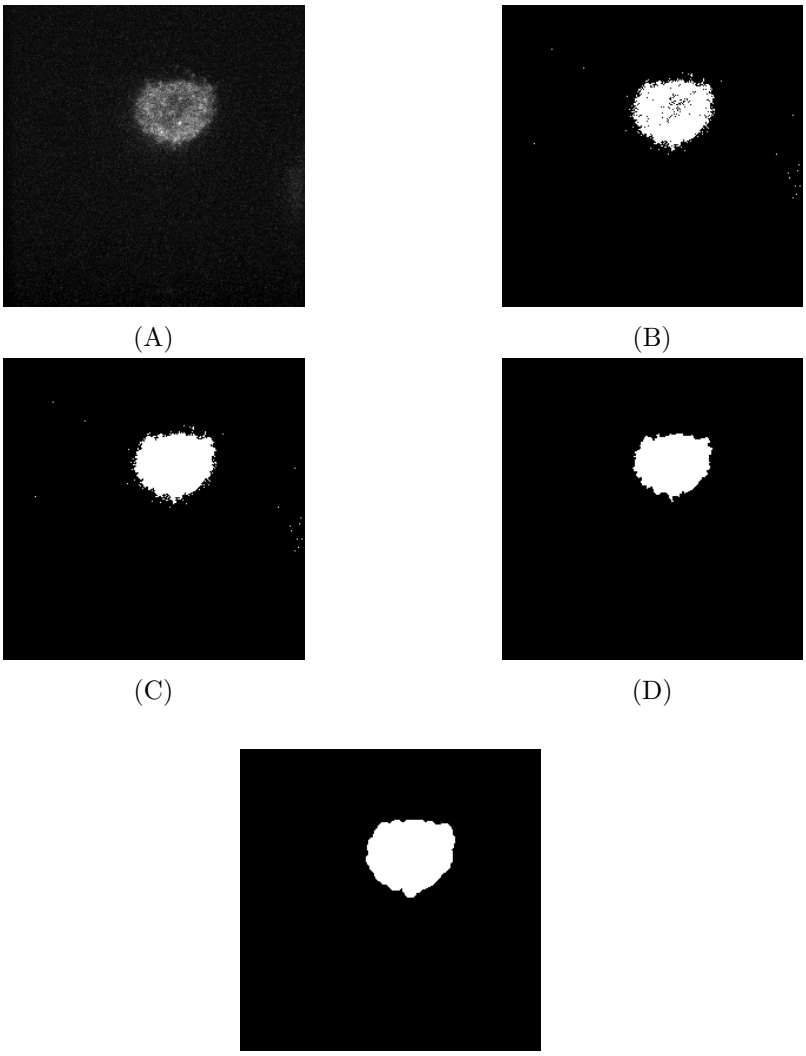



**Figure 5.3:** Binary Mask method: (A) Original Image. (B) After the Otsu algorithm has been applied to make a binary image. (C) After a fill holes algorithms has been applied. (D) After pixel erosion. (E) The final binary mask after dilation.

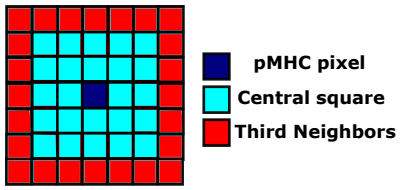
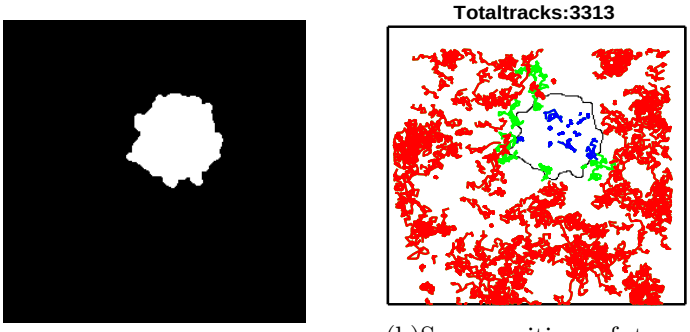



**Figure 5.4:** Sketch of intensity analysis of bulk-labelled molecules. Central pixel is a pMHC in the slow state. Ratio of average intensity of cyan+blue area to average intensity of red area is calculated for consecutive frames in which the pMHC is in the slow state.




(a) Binary cell mask.

(b)Superposition of trajectories over cell mask.

**Figure 5.5:** Processed images in each channel. (a) Binary mask used to localize the tracks outside (red), crossing boundary (green) and inside (blue) the cell-contact area. (b) Superposition of those tracks over the cell-contact area mask.

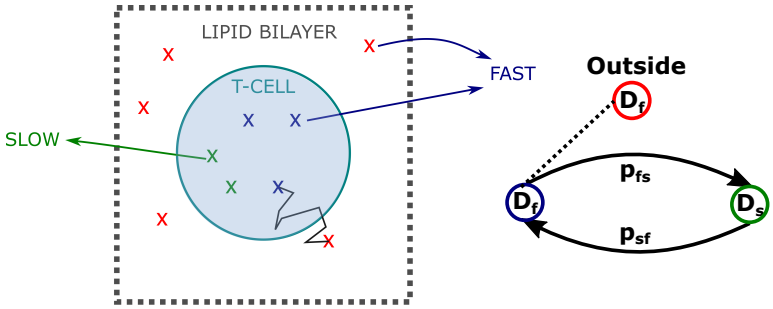
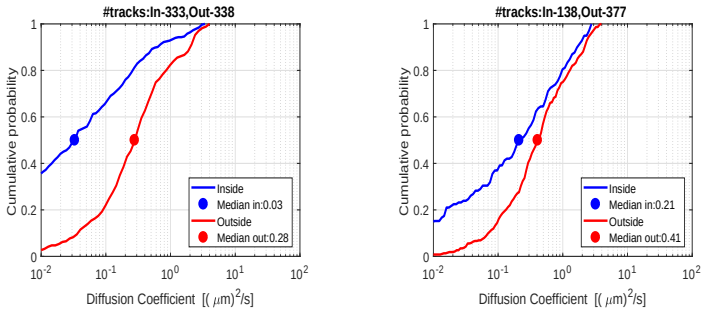


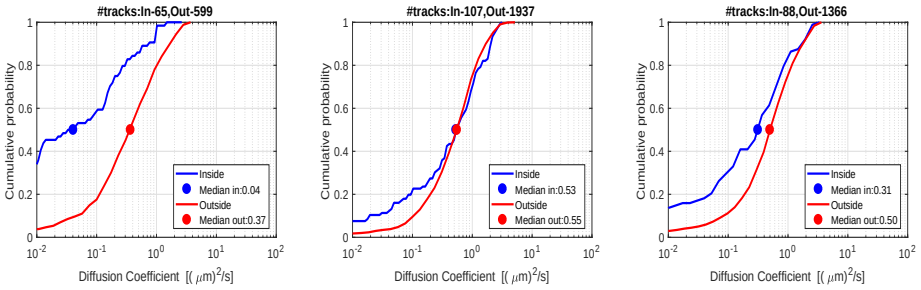

**Figure 5.6:** Sketch of the constrained 2-state model. The labelled proteins can be either in a fast or slow state when within the cell-contact area (blue and green x). When they are out of the cell-contact area, they can only be in the fast state.




WT MCC T102S MCC

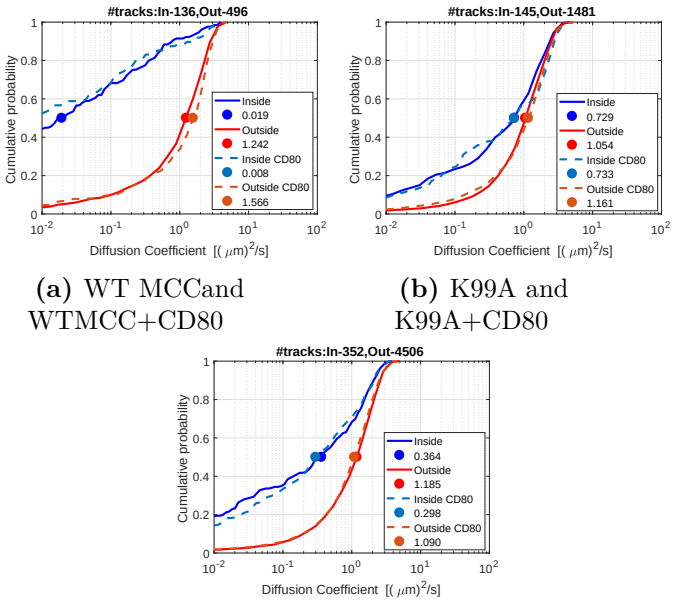
K99A MCC $\beta$2m MCC

**Figure 5.7:** pMHC mobility within the supported bilayer is modulated by interactions with AND T cells. Representative experiments are shown for four different pMHC: WT MCC, T102S MCC, K99A MCC and $\beta$2m. Diffusion coefficient cumulative distribution functions are plotted for tracks that are entirely outside cell contact areas (red) and entirely within a cell contact area (blue).




WT MCC T102S MCC K5 MCC

**Figure 5.8:** pMHC mobility within the supported bilayer is modulated by interactions with 5cc7 T cells. Representative experiments are shown for four different pMHC: WT MCC, T102S MCC, K99A MCC and $\beta$2m. Diffusion coefficient cumulative distribution functions are plotted for tracks that are entirely outside cell contact areas (red) and entirely within a cell contact area (blue).

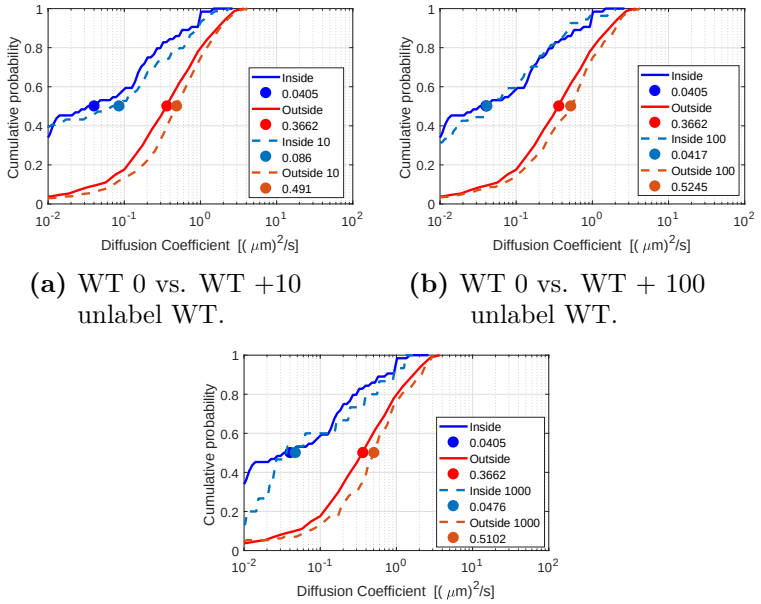



**(a)** WT MCCand WTMCC+CD80

**(b)** K99A and K99A+CD80

**(c)** T102S and T102S+CD80s

**Figure 5.9:** (pMHC mobility within the supported bilayer is modulated by interactions with AND T cells. Representative experiments are shown for three different pMHC: WT MCC, T102S MCC and K99A MCC. Two experimental conditions are shown: no addition of unlabelled CD80, addition of unlabelled CD80 for each of the three pMHC. Diffusion coefficient cumulative distribution functions are plotted for tracks that are entirely outside cell contact areas (red) and entirely within a cell contact area (blue). The dashed lines are for the experiments where we added CD80 to the bilayer. No significant difference in the mobility between experiments with and without CD80 was observed.

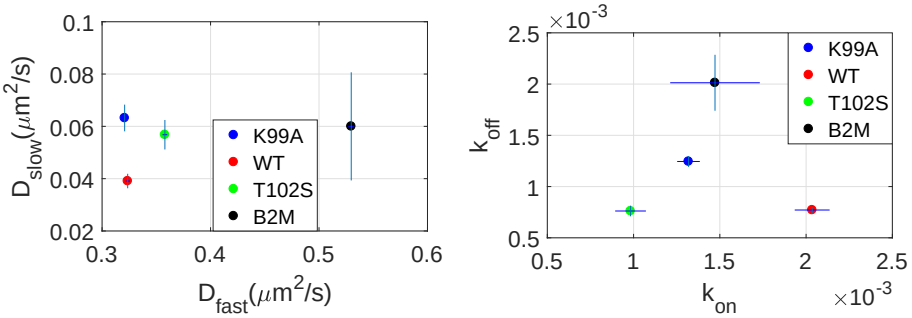



**(a)** WT 0 vs. WT +10 unlabel WT.

**(b)** WT 0 vs. WT + 100 unlabel WT.

**(c)** WT vs. WT + 1000 unlabel WT.

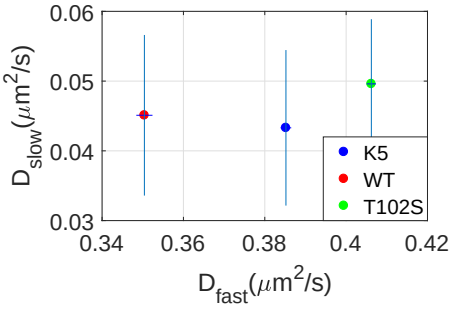
**Figure 5.10:** pMHC mobility within the supported bilayer is modulated by interactions with T cells. Representative experiments are shown for WT MCC pMHC. Four experimental conditions are shown: no addition of unlabelled peptide, addition of 10 unlabelled WT MCC pMHC over $\mu m^2$, addition of 100 unlabelled WT MCC pMHC over $\mu m^2$, and addition of 1000 unlabelled WT MCC pMHC over $\mu m^2$. These numbers indicate the concentration of unlabelled peptides. Diffusion coefficient cumulative distribution functions are plotted for tracks that are entirely outside cell contact areas (red) and entirely within a cell contact area (blue). The dashed lines are for the experiments where we added unlabelled WT MCC pMHCs to the bilayer. No significant difference in the mobility between experiments with and without unlabelled WT MCC pMHC was observed.



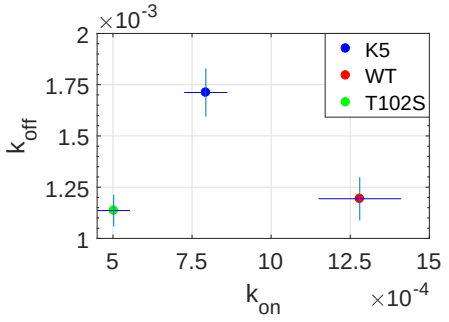

(a)AND TCRs: diffusivity




(b) AND TCRs: transition rate

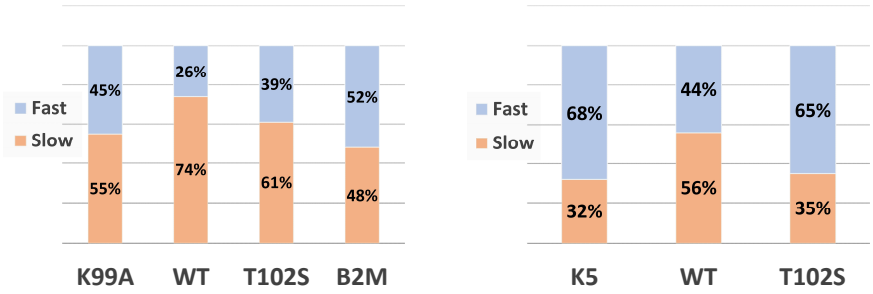



(c)5cc7 TCRs: diffusivity




(d) 5cc7 TCRs: transition rate

**Figure 5.11:** Constrained 2-state HMM results for seven different experiments. Four experiments were done with AND TCRs and four different peptide-MHCs: K99A, WT, T102S, and $\beta$2m. Tne other 3 used 5cc7 TCRs and 3 different peptide-MHCS: K5,WT and T102S.(a)-(c) diffusivities of slow and fast state for each one of the experiments, and (b)-(d) on and off rates, where $k_{\text{on}}$ is the transition rate from fast to slow, and $k_{\text{off}}$ is the transition rate from slow to fast.




(a) AND TCRs (b) 5cc7 TCRs

**Figure 5.12:** Constrained hidden Markov modeling of pMHC motion. Segmentation of states plots showing the percentage of slow (orange) and fast (blue) state for each experiments.

# CHAPTER 6

# Conclusions

This thesis described the development of new methods to analyze single-particle trajectories, and their application to experimental data. In Chapters 2-4, we focused on two questions related to single-particle tracking analysis. The first question regards experimental errors, their inclusion on the modelling, and how they impact on the accuracy of fitting. The second question is related to model size inference. In Chapter 5, we developed a method specific to a two colour experimental setting, and described a pipeline that one can use to analyse similar experimental data.

In Chapter 2, we described the two-state hidden Markov model taking into account experimental errors. This novel model is a combination of two previous methods. As we discussed in Chapter 2, the inclusion of experimental errors in the inference improves the accuracy of the estimates. Instead of the describing the displacements as a Brownian process, they are described as a measurement process, where the errors are added to the Brownian process. Under the assumption that the model for the experimental errors is correct, we showed that the diffusivity estimates for simulated data using our method are more accurate compared to the comparable estimates

when using the two-state hidden Markov model without taking into account experimental errors.

When modelling a biological system, we need to make some assumptions to simplify the system. In the process of choosing an assumption, one needs to weigh in the pros and cons. For example, modelling tracks as a pure Brownian motion is simple and requires the estimation of only one parameter. Additionally, it is a reasonable choice of process, because of the microscopic particle size and environment. However, the lack of heterogeneity of the model might not sufficiently explain the behaviour of the tracked molecule on the cell. The cell membrane is a very crowded environment, and a natural question to ask is whether the interactions of proteins on the cell membrane with this crowded environment is strong enough to affect their mobility. The simplicity of only one parameter for the one state pure diffusion fails to capture these effects. Instead, it just averages the possible effect of the crowding without giving any more details. Increasing the number of states allows us to add some complexity and, therefore, heterogeneity into our understanding of track mobility. Then, we have some quantitative measurement of the effects of these possible interactions. This is an example of the underfitting and overfitting dilemma. For a smaller number of states, we have a simpler model, but it might not capture all information from the data. As we increase the number of states, we acquire a more detailed picture of the mobility. However, we get to a point where the complexity is so high that it cannot be meaningfully fit to the available data. A helpful example is to imagine the number of states to be equal to the size of the data. More importantly, we do need to decide the number of states that best represents the experimental system. The number of states that best explains the mobility of a protein on the cellular membrane is of great interest to biology, because could give some insight on the organization of the cellular membrane.

The usual fitting procedure is to maximize the likelihood of a model and their parameters given the data. Then, one needs a priori number of states to calculate the likelihood. While we could calculate the likelihood for different models with different numbers of states, and then use some post-hoc

information criteria to decide the best model, here we developed a framework that fits the number of states and their parameters simultaneously to the data, effectively allowing the data to decide the best model.

In Chapter 3, we described the algorithm that fits the number of states as well, the infinite hidden Markov model for single-particle trajectories. We validate the algorithm using simulated data, and on experimental data. Comparing the different results, we observed the importance of fitting the number of states as well, since we consistently obtained a number of states larger than two.

We wanted to include the experimental errors in the estimation of the number of states so that the estimates are more accurate, similar to Chapter 2 for the two state model. Therefore, we implemented the SPT-$\infty$E algorithm in Chapter 4. Using simulated data, we conclude that the estimation process becomes more accurate once experimental errors are taking into consideration. We then follow to re-study the same set of experimental data. Our algorithm successfully recovered the parameters and number of states for simulated datasets when enough data were provided. However, the SPT-$\infty$E fails to recover the covariance matrices accurately when the data are not large enough. We fund that this effect is more noticeable when the data has a larger number of states.

There are many other approaches to analyse single-particle tracking mobility. Some methods classify the motion of the trajectory between two motions: a fast motion a and slow motion [9, 24], or between confined motion and pure diffusive motion using a hidden Markov framework [121], while estimating the confined potential parameters as well. There are machine learning algorithms that classify the motion for each track or for each displacement in each track, usually choosing between some sort of anomalous diffusion, pure diffusion and directed motion [49, 62, 84, 95]. More similar to our approach, there are methods that use information criteria to decide what is the best model between a one-state diffusive model or a 2-state model [122], a variational Bayes approach to select the best number of diffusive states and to estimate the diffusion coefficient taking into consideration experimental errors [67], a perturbation-expectation maximization analysis

that sorts segments of population trajectories into diffusive states (each segment comprises 15 frames of a trajectory) [102], and a method that uses non-parametric Bayesian to fit a Gaussian mixture model to the data [58]. In the latter method, each state is described by a Gaussian distribution, and the number of such distributions in the mixture is equivalent to the number of Markov states in the SPT-$\infty$ model.

For most of these methods above, the number of diffusive states is predefined. In comparison to them, our method is an advantage given that it also estimates the optimal number of states. For methods using an information criteria, we would first need to fit to some number of specific models, each of which would effectively be conditioned on its own correctness. It is not possible to compare the likelihoods of such individual models directly, and the uncertainty (variance in the parameter estimates) is also underestimated because our lack of knowledge about which model is correct is not included. The SPT-$\infty$ naturally incorporates the uncertainty due to model selection, while information criteria approaches are rather different and do not capture the uncertainty beyond a correction for the number of additional parameters in the more-complex model [3, 44]. Of course, information criteria have an important role to play when computational efficiency becomes important, as well as allowing model comparison when the possible models do not fit neatly together. Finally, the work by Karslake et al. [58] provides an alternative to our models, given that they also use a non-parametric Bayesian framework. The main difference is that they consider a Gaussian mixture model, instead of an infinite hidden Markov model, therefore transitions between states are not accounted for.

There are several possible future extensions of this work considering ways to further improve the current approach, or use a different approach to the number of states. Below, we briefly explain them.

- For the infinite hidden Markov model, we assume that transitions between states cannot happen during image capture. Thus, for an image stack with $N$ frames, we have the $N-1$ time intervals between frames where transitions between states can occur. Therefore, the time is dis-

cretized, and we have a discrete time Markov chain. This assumption is reasonable for the case where image acquisition is much faster than the rate of transitions between states. However, for a system with fast kinetics, one might need to further adapt this model to taken into account the possibility of transitions happening in between the image acquisition times. For that, we would need to use a continuous time Markov process. This would allow us to estimate the time a transition occurs and to which state the particle transits, as opposed to only sampling the sequence of states. One way to approach that is through the use of jump processes. Jump processes are stochastic processes that transition between states at random times. Thus, the time when a transition occurs would also be a random variable in the model, as opposed to the SPT-$\infty$ where the transitions occurs only on discrete fixed time points. Briefly, an extra process would be added to the framework, where the time when the transitions occurs are sampled from a specified distribution. Then, the state of each transition would be sampled, resulting in the sequence of states. Next, the parameters of the model would be sample conditioned on the data and on the sequence of states [59].

- Another way to further improve the SPT-$\infty$E is to consider other emission models rather than only Brownian motion (measurement process for the experimental errors). For example, one could allow for transient confinement of the particle within a potential well to resemble particle interactions [121].

- Following a different approach, one could develop a machine learning algorithm (a classification algorithm), such that the algorithm would classify the displacements between possible mode of motions. There have been some works in this area using neural nets, decision trees, random forests, and so on [29, 49, 62, 84]. A more challenging work though is to come up with a machine learning algorithm that, like the infinite hidden Markov, does not have a pre-defined number of classification groups. Therefore, there is no specific output size for

the machine to predict, and one would need to create a machine that learns the number of classification groups, so that another machine can be built to classify the data among the groups.

In Chapter 5, we developed an analysis pipeline for two-colour TIRF microscopy data to investigate the dynamics of peptide-MHC in a supported lipid bilayer with a T cell placed on top of it. We wanted to elucidate any correlations between the pMHC mobility and the T cell presence. Thus, we applied two models to quantify the pMHC mobility and its heterogeneity. Moreover, we developed a method to quantify the pixel intensity change on the vicinity of a pMHC in search of correlations between the pMHC mobility and the protein labelled at high density (TCR or CD45). This pipeline could be used in similar experimental settings, where one has a low density labelled molecule in one colour, whose mobility one wants to investigate, and a high density labelled molecule, which can be used to localize the cell.

Following a step-by-step procedure, we started with two existing models and combined them into the novel SPT-2E algorithm, with the goal of improving the fit of a two-state Markov model to experimental data. Next, we developed the SPT-$\infty$ algorithm to also estimate the number of diffusive states. To incorporate the experimental errors in the estimation process, we developed the SPT-$\infty$E algorithm. Finally, we created an analysis pipeline for two-colour TIRF microscopy, that includes a constrained two-state Markov model. Overall, this thesis shows the development of novel methods that substantially advance our ability to analyze single particle tracks. In addition, it shows how mathematical models can be powerful tools for the analysis and understanding of biological systems.

# Bibliography


[1] L. Abraham, H. Y. Lu, R. C. Falcão, J. Scurll, T. Jou, B. Irwin, R. Tafteh, M. R. Gold, and D. Coombs. Limitations of Qdot labelling compared to directly-conjugated probes for single particle tracking of B cell receptor mobility. *Scientific reports*, 7(1):11379, September 2017. {pages vi, xi, 2, 16, 17, 18, 32, 60, 61, 62, 63, 65, 68, 69, 70, 73, 104, 105, 106, 108, 109, 112, 134, 137, 141, 146, and 159.}

[2] E. Ambrose. A surface contact microscope for the study of cell movements. *Nature*, 178(4543):1194–1194, 1956. {pages 3, 7, and 8.}

[3] D. R. Anderson and K. P. Burnham. *Model selection and multi-model inference*. Springer, New York, 2nd edition, 2002. {pages 111 and 184.}

[4] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966. {pages 29 and 50.}

[5] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden Markov model. *Advances in neural information processing systems*, 1: 577–584, 2002. {pages 73 and 87.}

[6] A. J. Berglund. Statistics of camera-based single-particle tracking. *Physical review E*, 82:011917, 2010. doi:10.1103/PhysRevE.82.011917. {pages 31, 32, 34, 36, 37, 39, 71, 102, 112, 155, and 158.}

[7] J. Bernstein and J. Fricks. Analysis of single particle diffusion with transient binding using particle filtering. *J Theor Biol*, 401:109–21, 07 2016. doi:10.1016/j.jtbi.2016.04.013. {pages 72 and 112.}

[8] E. Betzig and R. J. Chichester. Single molecules observed by near-field scanning optical microscopy. *Science*, pages 1422–1425, 1993. {page 3.}

[9] P. J. Bosch, J. S. Kanger, and V. Subramaniam. Classification of dynamical diffusion states in single molecule tracking microscopy. *Biophys J*, 107(3):588–598, Aug 2014. doi:10.1016/j.bpj.2014.05.049. {pages 72 and 183.}

[10] P. C. Bressloff. *Diffusion in Cells: Random Walks and Brownian Motion*, pages 35–102. Springer International Publishing, Cham, 2014. ISBN 978-3-319-08488-6. doi:10.1007/978-3-319-08488-6_2. URL https://doi.org/10.1007/978-3-319-08488-6_2. {pages 21, 24, and 29.}

[11] Z. Bu and D. J. Callaway. Chapter 5 - Proteins move! protein dynamics and long-range allostery in cell signaling. In R. Donev, editor, *Protein Structure and Diseases*, volume 83 of *Advances in Protein Chemistry and Structural Biology*, pages 163–221. Academic Press, 2011. doi:https://doi.org/10.1016/B978-0-12-381262-9.00005-7. {page 1.}

[12] C. W. Cairo, R. Mirchev, and D. E. Golan. Cytoskeletal regulation couples LFA-1 conformational changes to receptor lateral mobility and clustering. *Immunity*, 25(2):297–308, 2006. {page 52.}

[13] C. W. Cairo, R. Das, A. Albohy, Q. J. Baca, D. Pradhan, J. S. Morrow, D. Coombs, and D. E. Golan. Dynamic regulation of CD45 lateral mobility by the spectrin-ankyrin cytoskeleton of T cells. *J Biol Chem*, 285 (15):11392–401, Apr 2010. doi:10.1074/jbc.M109.075648. {page 72.}

[14] G. Campi, R. Varma, and M. L. Dustin. Actin and agonist MHC-peptide complex-dependent T cell receptor microclusters as scaffolds for signaling. *J Exp Med*, 202(8):1031–6, Oct 2005. doi:10.1084/jem.20051182. {pages 164 and 170.}

[15] A. K. Chakraborty and A. Weiss. Insights into the initiation of TCR signaling. *Nature immunology*, 15(9):798, 2014. {page 150.}

[16] N. Chenouard, I. Bloch, and J.-C. Olivo-Marin. Multiple hypothesis tracking in cluttered condition. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 3621–3624. IEEE, 2009. {pages 14 and 15.}

[17] N. Chenouard, I. Bloch, and J.-C. Olivo-Marin. Multiple hypothesis tracking for cluttered biological image sequences. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2736–3750, 2013. {pages 14 and 15.}

[18] N. Chenouard, I. Smal, F. de Chaumont, M. Maška, I. F. Sbalzarini, Y. Gong, J. Cardinale, C. Carthel, S. Coraluppi, M. Winter, A. R. Cohen, W. J. Godinez, K. Rohr, Y. Kalaidzidis, L. Liang, J. Duncan, H. Shen, Y. Xu, K. E. G. Magnusson, J. Jaldén, H. M. Blau, P. Paul-Gilloteaux, P. Roudot, C. Kervrann, F. Waharte, J.-Y. Tinevez, S. L. Shorte, J. Willemse, K. Celler, G. P. van Wezel, H.-W. Dan, Y.-S. Tsai, C. Ortiz de Solórzano, J.-C. Olivo-Marin, and E. Meijering. Objective comparison of particle tracking methods. *Nat Methods*, 11(3):281–9, Mar 2014. doi:10.1038/nmeth.2808. {pages 10, 13, 14, 15, 16, 71, and 103.}

[19] A. G. Cherstvy, S. Thapa, C. E. Wagner, and R. Metzler. Non-Gaussian, non-ergodic, and non-Fickian diffusion of tracers in mucin hydrogels. *Soft matter*, 15(12):2526–2551, 2019. {page 72.}

[20] K. Choudhuri, D. Wiseman, M. H. Brown, K. Gould, and P. A. van der Merwe. T-cell receptor triggering is critically dependent on the dimensions of its peptide-MHC ligand. *Nature*, 436(7050):578–582, 2005. {pages 164 and 170.}

[21] E. Corse, R. A. Gottschalk, M. Krogsgaard, and J. P. Allison. Attenuated T cell responses to a high-potency ligand in vivo. *PLoS biology*, 8(9):e1000481, 2010. {pages 163 and 170.}

[22] J. C. Crocker and D. G. Grier. Methods of digital video microscopy for colloidal studies. *Journal of colloid and interface science*, 179(1): 298–310, 1996. {pages 2 and 10.}

[23] Y. Cui, M. Yu, X. Yao, J. Xing, J. Lin, and X. Li. Single-particle tracking for the quantification of membrane protein dynamics in living plant cells. *Molecular plant*, 11(11):1315–1327, 2018. {page 2.}

[24] R. Das, C. W. Cairo, and D. Coombs. A hidden Markov model for single particle tracks quantifies dynamic interactions between LFA-1 and the actin cytoskeleton. *PLoS Comput Biol*, 5(11):e1000556, Nov 2009. doi:10.1371/journal.pcbi.1000556. {pages 27, 29, 31, 32, 34, 37, 48, 49, 50, 51, 52, 68, 72, 104, 155, 156, and 183.}

[25] A. M. Davey and S. K. Pierce. Intrinsic differences in the initiation of B cell receptor signaling favor responses of human IgG+ memory B cells over IgM+ naive B cells. *The journal of immunology*, 188(7): 3332–3341, 2012. {page 109.}

[26] M. De Brabander, R. Nuydens, H. Geerts, and C. Hopkins. Dynamic behavior of the transferrin receptor followed in living epidermoid carcinoma A431 cells with nanovid microscopy. *Cell motility and the cytoskeleton*, 9(1):30–47, 1988. {pages 2 and 3.}

[27] F. de Chaumont, S. Dallongeville, N. Chenouard, N. Hervé, S. Pop, T. Provoost, V. Meas-Yedid, P. Pankajakshan, T. Lecomte, Y. Le Montagner, T. Lagache, A. Dufour, and J.-C. Olivo-Marin. Icy: an open bioimage informatics platform for extended reproducible research. *Nat Methods*, 9(7):690–6, Jun 2012. doi:10.1038/nmeth.2075. {pages 14 and 103.}

[28] C. Dietrich, B. Yang, T. Fujiwara, A. Kusumi, and K. Jacobson. Relationship of lipid rafts to transient confinement zones detected by single particle tracking. *Biophysical journal*, 82(1):274–284, 2002. {pages 31 and 72.}

[29] P. Dosset, P. Rassam, L. Fernandez, C. Espenel, E. Rubinstein, E. Margeat, and P.-E. Milhiet. Automatic detection of diffusion modes within biological membranes using back-propagation neural network. *BMC bioinformatics*, 17(1):1–12, 2016. {page 185.}

[30] A. Einstein. *Investigations on the Theory of the Brownian Movement*. Courier Corporation, 1956. {pages 18 and 29.}

[31] D. Ernst and J. Köhler. Measuring a diffusion coefficient by single-particle tracking: statistical analysis of experimental mean squared displacement curves. *Physical Chemistry Chemical Physics*, 15(3): 845–849, 2013. {page 71.}

[32] D. Ernst, M. Hellmann, J. Köhler, and M. Weiss. Fractional Brownian motion in crowded fluids. *Soft matter*, 8(18):4886–4889, 2012. {page 26.}

[33] J. H. Esensten, Y. A. Helou, G. Chopra, A. Weiss, and J. A. Bluestone. CD28 costimulation: From mechanism to therapy. *Immunity*, 44(5): 973–88, 05 2016. doi:10.1016/j.immuni.2016.04.020. {page 161.}

[34] R. C. Falcão and D. Coombs. Diffusion analysis of single particle trajectories in a Bayesian nonparametrics framework. *Physical biology*, 17(2):025001, 2020. {pages vi and 137.}

[35] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973. {pages 73, 74, 81, 82, 83, and 84.}

[36] D. Fink. A compendium of conjugate priors. See http://www. people. cornell. edu/pages/df36/CONJINTRnew% 20TEX. pdf, 1997. {page 118.}

[37] K. N. Fish. Total internal reflection fluorescence (TIRF) microscopy. *Current protocols in cytometry*, 50(1):12–18, 2009. {pages 3, 7, 8, and 9.}

[38] B. A. Frigyik, A. Kapila, and M. R. Gupta. Introduction to the Dirichlet distribution and related processes. Technical report, Department of Electrical Engineering, University of Washignton, UWEETR-2010-0006, 2010. {pages 74, 75, 78, 79, 80, 87, 203, and 206.}

[39] C. W. Gardiner et al. *Handbook of stochastic methods*, volume 3. springer Berlin, 1985. {pages 24, 26, and 29.}

[40] H. Geerts, M. De Brabander, R. Nuydens, S. Geuens, M. Moeremans, J. De Mey, and P. Hollenbeck. Nanovid tracking: a new automatic method for the study of mobility in living cells based on colloidal gold and video microscopy. *Biophysical journal*, 52(5):775–782, 1987. {pages 2 and 3.}

[41] J. Gelles, B. J. Schnapp, and M. P. Sheetz. Tracking kinesin-driven movements with nanometre-scale precision. *Nature*, 331(6155):450–453, 1988. {page 10.}

[42] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992. {page 94.}

[43] A. Gelman, G. Roberts, and W. Gilks. Efficient metropolis jumping hules. *Bayesian statistics*, 1996. {page 52.}

[44] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian data analysis*. CRC Press, Boca Raton, FL, 2013. {pages 111 and 184.}

[45] R. N. Germain. Computational analysis of T cell receptor signaling and ligand discrimination–past, present, and future. *FEBS letters*, 584 (24):4814–4822, 2010. {page 150.}

[46] J. F. Geweke et al. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. Technical report, Federal Reserve Bank of Minneapolis, 1991. {page 94.}

[47] A. Gnedin and S. Kerov. A characterization of GEM distributions. *Combinatorics, probability and computing*, 10(3):213–217, 2001. {page 83.}

[48] A. Grakoui, S. K. Bromley, C. Sumen, M. M. Davis, A. S. Shaw, P. M. Allen, and M. L. Dustin. The immunological synapse: a molecular machine controlling T cell activation. *Science*, 285(5425):221–227, 1999. {page 148.}

[49] N. Granik, L. E. Weiss, E. Nehme, M. Levin, M. Chein, E. Perlson, Y. Roichman, and Y. Shechtman. Single-particle diffusion characterization by deep learning. *Biophysical journal*, 117(2):185–192, 2019. {pages 183 and 185.}

[50] R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE transactions on pattern analysis and machine intelligence*, (4):532–550, 1987. {page 152.}

[51] M. Hirsch, R. Wareham, J. W. Yoon, D. J. Rolfe, L. C. Zanetti-Domingues, M. P. Hobson, P. J. Parker, M. L. Martin-Fernandez, and S. S. Singh. A global sampler of single particle tracking solutions for single molecule microscopy. *PloS one*, 14(10):e0221865, 2019. {pages 10 and 15.}

[52] R. Hooke. Milestones. *Philosophical transactions of the royal society B*, 135:143–145, 1845. {page 3.}

[53] S. Huda, B. Weigelin, K. Wolf, K. V. Tretiakov, K. Polev, G. Wilk, M. Iwasa, F. S. Emami, J. W. Narojczyk, M. Banaszak, et al. Lévy-like movement patterns of metastatic cancer cells revealed in microfabricated systems and implicated in vivo. *Nature communications*, 9(1): 1–11, 2018. {page 25.}

[54] J. B. Huppa, M. Axmann, M. A. Mörtelmaier, B. F. Lillemeier, E. W. Newell, M. Brameshuber, L. O. Klein, G. J. Schütz, and M. M. Davis. TCR–peptide–MHC interactions in situ show accelerated kinetics and increased affinity. *Nature*, 463(7283):963–967, 2010. {pages 163 and 170.}

[55] C. P. Igiri, Y. Singh, and D. Bhargava. An improved African buffalo optimization algorithm using chaotic map and chaotic-Lévy flight. *International Journal of Engineering & Technology*, 7(4):4570–4576, 2018. {page 25.}

[56] A. James, M. J. Plank, and A. M. Edwards. Assessing Lévy walks as models of animal foraging. *Journal of the royal society interface*, 8 (62):1233–1247, 2011. {page 26.}

[57] K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S. L. Schmid, and G. Danuser. Robust single-particle tracking in live-cell time-lapse sequences. *Nature methods*, 5(8):695–702, 2008. {pages 10 and 15.}

[58] J. D. Karslake, E. D. Donarski, S. A. Shelby, L. M. Demey, V. J. DiRita, S. L. Veatch, and J. S. Biteen. SMAUG: Analyzing single-molecule tracks with nonparametric Bayesian statistics. *Methods*, 193: 16–26, 2021. {pages 30, 102, and 184.}

[59] Z. Kilic, I. Sgouralis, and S. Pressé. Generalizing HMMs to continuous time for fast kinetics: hidden Markov jump processes. *Biophysical journal*, 120(3):409–423, 2021. {page 185.}

[60] P. K. Koo and S. G. J. Mochrie. Systems-level approach to uncovering diffusive states and their transitions from single-particle trajectories. *Phys Rev E*, 94(5-1):052412, Nov 2016. doi:10.1103/PhysRevE.94.052412. {page 112.}

[61] P. K. Koo, M. Weitzman, C. R. Sabanaygam, K. L. van Golen, and S. G. J. Mochrie. Extracting diffusive states of Rho GTPase in live cells: Towards in vivo biochemistry. *PLoS Comput Biol*, 11 (10):e1004297, Oct 2015. doi:10.1371/journal.pcbi.1004297. {pages 30 and 72.}

[62] P. Kowalek, H. Loch-Olszewska, and J. Szwabiński. Classification of diffusion modes in single-particle tracking data: Feature-based versus deep-learning approach. *Physical review E*, 100(3):032410, 2019. {pages 183 and 185.}

[63] D. Krapf. Mechanisms underlying anomalous diffusion in the plasma membrane. *Current topics in membranes*, 75:167–207, 2015. {page 26.}

[64] A. Kusumi, Y. Sako, and M. Yamamoto. Confined lateral diffusion of membrane receptors as studied by single particle tracking (nanovid microscopy). Effects of calcium-induced differentiation in cultured epithelial cells. *Biophysical journal*, 65(5):2021–2040, 1993. {pages 2 and 21.}

[65] K.-H. Lee, A. R. Dinner, C. Tu, G. Campi, S. Raychaudhuri, R. Varma, T. N. Sims, W. R. Burack, H. Wu, J. Wang, et al. The immunological synapse balances T cell receptor signaling and degradation. *Science*, 302(5648):1218–1222, 2003. {page 148.}

[66] D. A. Leon and A. Gonzalez. Mutations as Lévy flights. *Scientific reports*, 11(1):1–8, 2021. {page 25.}

[67] M. Lindén and J. Elf. Variational algorithms for analyzing noisy multistate diffusion trajectories. *Biophysical Journal*, 2018. {pages 30, 72, and 183.}

[68] M. Lindén, V. Ćurić, E. Amselem, and J. Elf. Pointwise error estimates in localization microscopy. *Nature communications*, 8(1):1–9, 2017. {page 102.}

[69] S. T. Low-Nam, K. A. Lidke, P. J. Cutler, R. C. Roovers, P. M. P. van Bergen en Henegouwen, B. S. Wilson, and D. S. Lidke. ErbB1 dimerization is promoted by domain co-confinement and stabilized by ligand binding. *Nat Struct Mol Biol*, 18(11):1244–9, Oct 2011. doi:10.1038/nsmb.2135. {page 72.}

[70] D. Luengo, L. Martino, M. Bugallo, V. Elvira, and S. Särkkä. A survey of Monte Carlo methods for parameter estimation. *EURASIP journal on advances in signal processing*, 2020:1–62, 2020. {page 52.}

[71] C. Manzo and M. F. Garcia-Parajo. A review of progress in single particle tracking: from methods to biophysical insights. *Reports on progress in physics*, 78(12):124601, 2015. {pages 1, 2, 3, 4, 5, 11, 12, 13, 16, 17, 25, and 26.}

[72] D. S. Martin, M. B. Forstner, and J. A. Käs. Apparent subdiffusion inherent to single particle tracking. *Biophysical journal*, 83(4):2109–2117, 2002. {page 21.}

[73] T. G. Mason, K. Ganesan, J. H. van Zanten, D. Wirtz, and S. C. Kuo. Particle tracking microrheology of complex fluids. *Physical review letters*, 79(17):3282, 1997. {page 2.}

[74] A. L. Mattheyses, S. M. Simon, and J. Z. Rappoport. Imaging with total internal reflection fluorescence microscopy for the cell biologist. *Journal of Cell Science*, 123(21):3621–3628, 2010. ISSN 0021-9533. doi:10.1242/jcs.056218. URL https://jcs.biologists.org/content/123/21/3621. {pages 3, 7, 8, and 9.}

[75] N. Meilhac, L. Le Guyader, L. Salome, and N. Destainville. Detection of confinement and jumps in single-molecule membrane trajectories. *Physical Review E*, 73(1):011915, 2006. {page 72.}

[76] F. A. Merchant, K. A. Bartels, A. C. Bovik, and K. R. Diller. 10.9 - Confocal microscopy. In A. BOVIK, editor, *Handbook of Image and Video Processing (Second Edition)*, Communications, Networking and Multimedia, pages 1291–XLI. Academic Press, Burlington, second edition edition, 2005. ISBN 978-0-12-119792-6. doi:https://doi.org/10.1016/B978-012119792-6/50135-2. URL https://www.sciencedirect.com/science/article/pii/B9780121197926501352. {pages 3 and 6.}

[77] R. Metzler and J. Klafter. The random walk's guide to anomalous diffusion: a fractional dynamics approach. *Physics reports*, 339(1):1–77, 2000. {pages 25 and 26.}

[78] R. Metzler, J.-H. Jeon, A. G. Cherstvy, and E. Barkai. Anomalous diffusion models and their properties: non-stationarity, non-ergodicity,

and ageing at the centenary of single particle tracking. *Physical chemistry chemical physics*, 16(44):24128–24164, 2014. {pages 25 and 26.}

[79] X. Michalet. Mean square displacement analysis of single-particle trajectories with localization error: Brownian motion in an isotropic medium. *Physical review E*, 82(4):041914, 2010. {pages 31 and 71.}

[80] A. Mikš and J. Novák. Point spread function of an optical system with defocus and spherical aberration—analytical formulas. *Applied optics*, 58(21):5823–5829, 2019. {page 12.}

[81] J. W. Miller and M. T. Harrison. A simple example of dirichlet process mixture inconsistency for the number of components. *arXiv preprint arXiv:1301.2708*, 2013. {page 96.}

[82] N. Monnier, Z. Barry, H. Y. Park, K.-C. Su, Z. Katz, B. P. English, A. Dey, K. Pan, I. M. Cheeseman, R. H. Singer, et al. Inferring transient particle transport dynamics in live cells. *Nature methods*, 12 (9):838, 2015. {page 72.}

[83] D. Montiel, H. Cang, and H. Yang. Quantitative characterization of changes in dynamical behavior for single-particle tracking studies. *The journal of physical chemistry B*, 110(40):19763–19770, 2006. {page 36.}

[84] G. Muñoz-Gil, M. A. Garcia-March, C. Manzo, J. D. Martín-Guerrero, and M. Lewenstein. Single trajectory characterization via machine learning. *New journal of physics*, 22(1):013010, 2020. {pages 183 and 185.}

[85] K. Murphy. Filtering, smoothing and the junction tree algorithm. *University of California, Berkeley Technical Report*, 1999. {page 51.}

[86] K. Murphy and C. Weaver. *Janeway's Immunobiology*. Garland Science, New York, 9th edition, 2017. {page 149.}

[87] K. P. Murphy. Switching Kalman filters. 1998. {page 51.}

[88] S. Niekamp, N. Stuurman, and R. D. Vale. A 6-nm ultra-photostable DNA FluoroCube for fluorescence imaging. *Nature methods*, 17(4): 437–441, 2020. {pages 5 and 6.}

[89] B. Nijboer. *The diffraction theory of aberrations.* PhD thesis, 1942. Relation: http://www.rug.nl/ Rights: Wolters. {page 12.}

[90] J.-C. Olivo-Marin. Extraction of spots in biological images using multiscale products. *Pattern recognition*, 35(9):1989–1996, 2002. {page 14.}

[91] F. Oswald, E. L. Bank, Y. J. Bollen, and E. J. Peterman. Imaging and quantification of trans-membrane protein diffusion in living bacteria. *Physical chemistry chemical physics*, 16(25):12625–12634, 2014. {page 31.}

[92] N. Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979. {page 151.}

[93] M. Ott, Y. Shai, and G. Haran. Single-particle tracking reveals switching of the HIV fusion peptide between two diffusive modes in membranes. *The Journal of Physical Chemistry B*, 117(42):13308–13321, 2013. {page 72.}

[94] F. Persson, M. Lindén, C. Unoson, and J. Elf. Extracting intracellular diffusive states and transition rates from single-molecule tracking data. *Nature methods*, 10(3):265–269, 2013. {page 30.}

[95] H. D. Pinholt, S. S.-R. Bohr, J. F. Iversen, W. Boomsma, and N. S. Hatzakis. Single-particle diffusional fingerprinting: A machine-learning framework for quantitative analysis of heterogeneous diffusion. *Proceedings of the national academy of sciences*, 118(31), 2021. {page 183.}

[96] V. Prasad, D. Semwogerere, and E. R. Weeks. Confocal microscopy of colloids. *Journal of physics: condensed matter*, 19(11):113102, 2007. {pages 3, 4, and 6.}

[97] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical recipes: the art of scientific computing: Fortran version*, volume 702. Cambridge University Press Cambridge, UK, 1986. {page 50.}

[98] E. M. Purcell. Life at low Reynolds number. *American journal of physics*, 45(1):3–11, 1977. {page 22.}

[99] H. Qian, M. P. Sheetz, and E. L. Elson. Single particle tracking. analysis of diffusion and flow in two-dimensional systems. *Biophysical journal*, 60(4):910–921, 1991. {pages 21, 31, 36, and 71.}

[100] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989. ISSN 0018-9219. doi:10.1109/5.18626. {pages 52, 87, 93, and 156.}

[101] P. K. Relich, M. J. Olah, P. J. Cutler, and K. A. Lidke. Estimation of the diffusion constant from intermittent trajectories with variable position uncertainties. *Physical Review E*, 93(4):042401, 2016. {pages 71 and 102.}

[102] I. Rey-Suarez, B. A. Wheatley, P. Koo, A. Bhanja, Z. Shu, S. Mochrie, W. Song, H. Shroff, and A. Upadhyaya. WASP family proteins regulate the mobility of the B cell receptor during signaling activation. *Nature communications*, 11, 2020. {pages 112, 113, and 184.}

[103] A. Reynolds. On the anomalous diffusion characteristics of membrane-bound proteins. *Physics letters A*, 342(5-6):439–442, 2005. {page 26.}

[104] A. M. Reynolds and C. J. Rhodes. The Lévy flight paradigm: random search patterns and mechanisms. *Ecology*, 90(4):877–887, 2009. {page 26.}

[105] N. Rusk. The fluorescence microscope. *Nature cell biology*, 11(1): S8–S9, 2009. {pages 4 and 6.}

[106] N. Ruthardt, D. C. Lamb, and C. Bräuchle. Single-particle tracking as a quantitative microscopy-based approach to unravel cell entry mechanisms of viruses and pharmaceutical nanoparticles. *Molecular therapy*, 19(7):1199–1211, 2011. {page 2.}

[107] Y. Sako and A. Kusumi. Compartmentalized structure of the plasma membrane for receptor movements as revealed by a nanometer-level motion analysis. *Journal of cell biology*, 125(6):1251–1264, 1994. {page 21.}

[108] D. K. Sasmal, W. Feng, S. Roy, P. Leung, Y. He, C. Cai, G. Cao, H. Lian, J. Qin, E. Hui, et al. TCR–pMHC bond conformation controls TCR ligand discrimination. *Cellular & molecular immunology*, 17(3): 203–217, 2019. {pages 150, 159, 163, 164, and 170.}

[109] M. J. Saxton. Lateral diffusion in an archipelago. single-particle diffusion. *Biophysical journal*, 64(6):1766–1780, 1993. {page 21.}

[110] M. J. Saxton. Single-particle tracking: connecting the dots. *Nature methods*, 5(8):671–672, 2008. {pages 2, 10, 11, and 13.}

[111] M. J. Saxton. A particle tracking meet. *Nature methods*, 11(3):247–248, 2014. {page 13.}

[112] M. J. Saxton and K. Jacobson. Single-particle tracking: applications to membrane dynamics. *Annual review of biophysics and biomolecular structure*, 26(1):373–399, 1997. {pages 2, 21, and 71.}

[113] C. Selle, F. Rückerl, D. S. Martin, M. B. Forstner, and J. A. Käs. Measurement of diffusion in Langmuir monolayers by single-particle tracking. *Physical Chemistry Chemical Physics*, 6(24):5535–5542, 2004. {page 72.}

[114] A. Sergé, N. Bertaux, H. Rigneault, and D. Marguet. Dynamic multiple-target tracing to probe spatiotemporal cartography of cell membranes. *Nature methods*, 5(8):687–694, 2008. {pages 17 and 141.}

[115] I. Sgouralis and S. Pressé. An introduction to infinite HMMs for single-molecule data analysis. *Biophysical journal*, 112(10):2021–2029, 2017. {pages 73, 87, and 113.}

[116] I. Sgouralis and S. Pressé. ICON: an adaptation of infinite HMMs for time traces with drift. *Biophysical journal*, 112(10):2117–2126, 2017. {page 73.}

[117] H. Shen, L. J. Tauzin, R. Baiyasi, W. Wang, N. Moringo, B. Shuang, and C. F. Landes. Single particle tracking: From theory to biophysical applications. *Chem Rev*, 117(11):7331–7376, Jun 2017. doi:10.1021/acs.chemrev.6b00815. {pages 2 and 71.}

[118] L. V. Sibener, R. A. Fernandes, E. M. Kolawole, C. B. Carbone, F. Liu, D. McAffee, M. E. Birnbaum, X. Yang, L. F. Su, W. Yu, S. Dong, M. H. Gee, K. M. Jude, M. M. Davis, J. T. Groves, W. A. Goddard, 3rd, J. R. Heath, B. D. Evavold, R. D. Vale, and K. C. Garcia. Isolation of a structural mechanism for uncoupling T cell receptor signaling from peptide-MHC binding. *Cell*, 174(3):672–687.e27, Jul 2018. doi:10.1016/j.cell.2018.06.017. {pages 164 and 170.}

[119] J. P. Siebrasse, I. Djuric, U. Schulze, M. A. Schlüter, H. Pavenstädt, T. Weide, and U. Kubitscheck. Trajectories and single-particle tracking data of intracellular vesicles loaded with either SNAP-Crb3A or SNAP-Crb3B. *Data in brief*, 7:1665–1669, 2016. {page 2.}

[120] R. Simson, E. D. Sheets, and K. Jacobson. Detection of temporary lateral confinement of membrane proteins using single-particle tracking analysis. *Biophysical journal*, 69(3):989–993, 1995. {page 72.}

[121] P. J. Slator and N. J. Burroughs. A hidden Markov model for detecting confinement in single-particle tracking trajectories. *Biophysical journal*, 115(9):1741–1754, Nov 2018. doi:10.1016/j.bpj.2018.09.005. {pages 31, 72, 102, 112, 183, and 185.}

[122] P. J. Slator, C. W. Cairo, and N. J. Burroughs. Detection of diffusion heterogeneity in single particle tracking trajectories using a hidden Markov model with measurement noise propagation. *PLoS One*, 10 (10):e0140759, 2015. doi:10.1371/journal.pone.0140759. {pages 31, 72, and 183.}

[123] L. M. Sompayrac. *How the immune system works*. John Wiley & Sons, 2019. {page 1.}

[124] Y. W. Teh. Dirichlet process., 2010. {pages 74, 75, 76, 80, 83, 84, and 87.}

[125] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the american statistical association*, 101(476):1566–1581, 2006. {pages 73, 83, 84, 86, and 87.}

[126] S. Thapa, M. A. Lomholt, J. Krog, A. G. Cherstvy, and R. Metzler. Bayesian analysis of single-particle tracking data using the nested-sampling algorithm: maximum-likelihood model selection applied to stochastic-diffusivity data. *Physical Chemistry Chemical Physics*, 20 (46):29018–29037, 2018. {page 72.}

[127] J. E. Thaxton and Z. Li. To affinity and beyond: harnessing the T cell receptor for cancer immunotherapy. *Hum Vaccin Immunother*, 10 (11):3313–21, 2014. doi:10.4161/21645515.2014.973314. {page 150.}

[128] G. I. Valderrama-Bahamóndez and H. Fröhlich. MCMC techniques for parameter estimation of ODE based models in systems biology. *Frontiers in applied mathematics and statistics*, 5:55, 2019. {page 94.}

[129] P. A. Van Der Merwe and O. Dushek. Mechanisms for T cell receptor triggering. *Nature reviews immunology*, 11(1):47–55, 2011. {page 150.}

[130] J. Van Gael. *Bayesian Nonparametric Hidden Markov Models*. PhD thesis, University of Cambridge, 2011. {pages 31, 73, 74, 82, and 83.}

[131] J. Van Gael, Y. Saatci, Y. W. Teh, and Z. Ghahramani. Beam sampling for the infinite hidden Markov model. In *Proceedings of the 25th international conference on Machine learning*, pages 1088–1095, 2008. {pages 73, 87, and 92.}

[132] J. Vangindertael, R. Camacho, W. Sempels, H. Mizuno, P. Dedecker, and K. Janssen. An introduction to optical super-resolution microscopy for the adventurous biologist. *Methods and applications in fluorescence*, 6(2):022003, 2018. {pages 6, 11, 12, and 16.}

[133] R. Varma, G. Campi, T. Yokosuka, T. Saito, and M. L. Dustin. T cell receptor-proximal signals are sustained in peripheral microclusters and terminated in the central supramolecular activation cluster. *Immunity*, 25(1):117–27, Jul 2006. doi:10.1016/j.immuni.2006.04.010. {page 164.}

[134] R. Varma, G. Campi, T. Yokosuka, T. Saito, and M. L. Dustin. T cell receptor-proximal signals are sustained in peripheral microclusters and terminated in the central supramolecular activation cluster. *Immunity*, 25(1):117–127, 2006. {pages 148, 164, and 170.}

[135] C. L. Vestergaard, P. C. Blainey, and H. Flyvbjerg. Optimal estimation of diffusion coefficients from single-particle trajectories. *Physical review E*, 89(2):022726, 2014. {pages 31, 71, and 102.}

[136] C. L. Vestergaard, P. C. Blainey, and H. Flyvbjerg. Single-particle trajectories reveal two-state diffusion-kinetics of hOGG1 proteins on DNA. *Nucleic acids research*, 46(5):2446–2458, 2018. {pages 31 and 72.}

[137] S. G. Walker. Sampling the Dirichlet mixture model with slices. *Communications in statistics—simulation and computation*, 36(1):45–54, 2007. {page 92.}

[138] D. Weihs, T. G. Mason, and M. A. Teitell. Bio-microrheology: a frontier in microrheology. *Biophysical journal*, 91(11):4296–4305, 2006. {page 2.}

[139] D. Wirtz. Particle-tracking microrheology of living cells: principles and applications. *Annual review of biophysics*, 38:301–326, 2009. {page 2.}

[140] J. Wishart. The generalised product moment distribution in samples from a normal multivariate population. *Biometrika*, 20(1/2):32–52, 1928. {page 118.}

[141] E. Wolf. The diffraction theory of aberrations. *Reports on progress in physics*, 14(1):95, 1951. {page 12.}

[142] M. Yanagawa, M. Hiroshima, Y. Togashi, M. Abe, T. Yamashita, Y. Shichida, M. Murata, M. Ueda, and Y. Sako. Single-molecule diffusion-based estimation of ligand effects on G protein-coupled receptors. *bioRxiv*, page 205161, 2018. {page 72.}

[143] M. U. Zahid, L. Ma, S. J. Lim, and A. M. Smith. Single quantum dot tracking reveals the impact of nanoparticle surface on intracellular state. *Nature communications*, 9(1):1–11, 2018. {page 2.}

[144] F. Zhang, G. M. Lee, and K. Jacobson. Protein lateral mobility as a reflection of membrane microstructure. *Bioessays*, 15(9):579–588, 1993. {page 2.}

# APPENDIX A

## Demonstration of the neutrality property of Dirichlet distributions

This appendix shows a demonstration of the neutrality property of Dirichlet distributions. This derivation is based on the work by Maya et al. [38].

### Neutrality

Suppose that $\boldsymbol{A} \sim Dir(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha}$, and $\boldsymbol{A}$ are vectors of length m. Moreover, $\boldsymbol{A}$ is defined over the (m-1)-dimensional probability simplex, i.e. $\sum_i^m A_i = 1$, and $0 < A_i < 1$ for $0 < i < m$. We want to show that $A_j$ is independent of the vector formed by the remaining elements normalized by their total sum.

Let's define a second vector $\boldsymbol{Y}$ such that $Y_i = \dfrac{A_i}{1 - A_j}$ for $i = 1, \ldots, j-2, j+1, \ldots, m$, $Y_{j-1} = 1 - \sum\limits_{\substack{i=1 \\ i \neq j-1,j}}^{m} Y_i$, and $Y_j = A_j$. Then $\sum\limits_{i \neq j}^{m} Y_i = 1$. Next, we find the transformation that maps $A_i$ to $Y_i$. Since we have the constraint $\sum_{i \neq j-1,j} Y_i = 1$, we must have $m-1$ variables on $\boldsymbol{Y}$, where $Y_{j-1}$ is a function of the rest. Then,

$$
\begin{aligned}
(A_1, \ldots, A_{j-2}, A_j, \ldots, A_m) = & \quad T\left(Y_1, \ldots, Y_{j-2}, Y_j, \ldots, Y_m\right), \\
= & \quad ((1-A_j)Y_1, \ldots, (1-A_j)Y_{j-2} \\
& \quad (1-A_j)Y_j, \ldots, (1-A_j)Y_m).
\end{aligned}
$$

Next, we apply the change of variables rule on the vector $\boldsymbol{Y}$, i.e. $p(\boldsymbol{Y};\boldsymbol{\alpha}) = (p(\boldsymbol{A};\boldsymbol{\alpha}) \circ T)(\boldsymbol{Y}) \times \mid det(J_T) \mid$, where

$$
p(\boldsymbol{A};\boldsymbol{\alpha}) = \quad \frac{\Gamma(\sum_i^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \left( \prod_{\substack{i=1 \\ i \neq j-1}}^{m} A_i^{\alpha_i - 1} \right) \left( 1 - \sum_{i \neq j-1} A_i \right)^{\alpha_{k-1}-1}, \quad \text{(A.1)}
$$

and $J_T$ is the Jacobian matrix of the transformation $T$, to get an expression for $p(\boldsymbol{Y};\boldsymbol{\alpha})$. The joint density of the new random variable is:

$$
\begin{aligned}
p(\boldsymbol{Y};\boldsymbol{\alpha}) = & \frac{\Gamma(\sum_i^k \alpha_i)}{\prod_{i=1}^m \Gamma(\alpha_i)} \left( \prod_{\substack{i=1 \\ i \neq j-1, j}}^{m} ((1-Y_j)Y_i)^{\alpha_i - 1} \right) \\
& \left( Y_j^{\alpha_j - 1} \right) \left( 1 - \sum_{\substack{i=1 \\ i \neq j-1, j}}^{m} Y_i(1-Y_j) - Y_j \right)^{\alpha_{j-1}-1} (1-Y_j)^{m-2},
\end{aligned}
$$

where $\mid det(J_T) \mid = (1-Y_j)^{m-2}$, where we define:

$$J_T = \begin{bmatrix} (1-Y_j) & 0 & \cdots & 0 & -Y_1 & 0 & 0 & 0 \\ 0 & (1-Y_j) & \cdots & 0 & -Y_2 & 0 & 0 & 0 \\ \vdots & 0 & \ddots & \vdots & \vdots & \vdots & 0 & 0 \\ 0 & \vdots & 0 & (1-Y_j) & -Y_{j-2} & 0 & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & 0 & Y_{j+1} & (1-Y_j) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & \ddots & 0 \\ 0 & 0 & 0 & \cdots & -Y_m & 0 & 0 & (1-Y_j) \end{bmatrix}.$$

We notice that:

$$\sum_{\substack{i=1 \\ i\neq j-1,j}}^{m} Y_i(1-Y_j) - Y_j = \left(1 - \sum_{i=1, i\neq j-1, j} Y_i\right)(1-Y_j) \tag{A.2}$$

$$= Y_{j-1}(1-Y_j) \tag{A.3}$$

Applying (A.3) to the expression for $p(\boldsymbol{Y};\boldsymbol{\alpha})$, and simplifying, we obtain:

$$p(\boldsymbol{Y};\boldsymbol{\alpha}) = \frac{\Gamma(\sum_i^k \alpha_i)}{\prod_{i=1}^m \Gamma(\alpha_i)} \left(\prod_{\substack{i=1 \\ i\neq j}}^{m} Y_i^{\alpha_i - 1}\right) Y_j^{\alpha_j - 1}(1-Y_j)^Z,$$

where $Z = \sum_{i=1, i\neq j}^{m} (\alpha_i - 1) + m - 2 = \sum_{i=1, i\neq j}^{m} \alpha_i - 1.$

We can rewrite (A.3) as:

$$p(\boldsymbol{Y};\boldsymbol{\alpha}) = \left( \frac{\Gamma(\sum_i^k \alpha_i)}{\Gamma(\alpha_j)\Gamma(\sum\limits_{\substack{i=1\\ i\neq j}}^{k} \alpha_i)} Y_j^{\alpha_j - 1}(1-Y_j)^{\sum\limits_{i=1, i\neq j}^{m} \alpha_i - 1} \right) \left( \frac{\Gamma(\sum\limits_{\substack{i=1\\ i\neq j}}^{k} \alpha_i)}{\prod\limits_{\substack{i=1\\ i\neq j}}^{m} \Gamma(\alpha_i)} \left( \prod_{\substack{i=1\\ i\neq j}}^{m} Y_i^{\alpha_i - 1} \right) \right). \tag{A.4}$$

One can see that the density factors into two independent terms. Rewriting, we have:

$$\begin{aligned} p(\boldsymbol{Y};\boldsymbol{\alpha}) =& p\left(Y_1, \ldots, Y_{j-1}, Y_{j+1}, \ldots, Y_m \mid \alpha_1, \ldots, \alpha_{j-1}, \alpha_{j+1}, \ldots, \alpha_m\right) \\ & p\left(Y_j \mid \alpha_j, \sum_{\substack{i=1^m\\ i\neq j}} \alpha_1\right), \\ =& p\left(\tfrac{A_1}{1-A_j}, \ldots, \tfrac{A_{j-1}}{1-A_j}, \tfrac{A_{j+1}}{1-A_j}, \ldots, \tfrac{A_m}{1-A_j} \mid \alpha_1, \ldots, \alpha_{j-1}, \alpha_{j+1}, \ldots, \alpha_m\right) \\ & p\left(A_j \mid \alpha_j, \sum_{\substack{i=1^m\\ i\neq j}} \alpha_i\right). \end{aligned}$$

We conclude that $A_j$ is independent of the vector $(\frac{A_1}{1-A_j}, \ldots, \frac{A_{j-1}}{1-A_j}, \frac{A_{j+1}}{1-A_j}, \ldots, \frac{A_m}{1-A_j})$ given $\boldsymbol{A} \sim Dir(\boldsymbol{\alpha})$. Finally, $(\frac{A_1}{1-A_j}, \ldots, \frac{A_{j-1}}{1-A_j}, \frac{A_{j+1}}{1-A_j}, \ldots, \frac{A_m}{1-A_j})$ follows a Dirichlet distribution with parameter $(\alpha_1, \ldots, \alpha_{j-1}, \alpha_{j+1}, \ldots, \alpha_m)$, and $A_j$ follows a Beta distribution with parameters $\alpha_j$, and $\sum\limits_{\substack{i=1^m\\ i\neq j}} \alpha_i$. [38]