LUND
UNIVERSITY

MASTER THESIS - 60 CREDITS

Simultaneous Bayesian parameter estimation and particle-tracking including calculation of mis-linking probabilities

---

Author:
Lennart GOLKS

Supervisor:
Tobias AMBJÖRNSSON

A Thesis submitted in Fulfillment of the Requirements
for the Degree of Master of Science
in the
Computational Biology and Biological Physics Group,
Department of Theoretical Physics in Lund

November 18, 2021

LUND UNIVERSITY

Abstract

Simultaneous Bayesian parameter estimation and particle-tracking including calculation of mis-linking probabilities

by Lennart GOLKS

Since 1994 super-resolution microscopes enable us to visualize processes in the nanometer regime where bio-molecules work. Consequently, there is a great need for methods analyzing the generated data to transfer the motion of molecules, seen as white dots, into trajectories. Important steps in understanding bio-molecular behavior are first the detection of those and then generating trajectories based on a physical model.

Many particle-tracking methods have been reviewed and it was concluded that it is advisable when linking dots into trajectories to know the particle dynamics [Chenouard et al., "Objective comparison of particle tracking methods" in Nature methods 16.5 2019, pp. 387-395]. However, this leads to a so-called catch-22 dilemma as the physical model describing the particles' motion is parameter dependent and so is the physical model-based linking process. To solve this dilemma we implement a Bayesian framework providing the best-fitting parameters and proposing trajectories in one go. This method does not require any prior information and is based on a parameter-dependent Brownian motion model with drift. In addition, we are the first to give mis-*linking* probabilities for each proposed step.

Our proposed method recovers trajectories well and estimates the diffusion constant and drift velocity of simulated data successfully. The calculation of mis-*linking* probabilities in unconstrained Brownian motion agrees with the ground truth recovery rate of the molecules' steps. We note that our methodology works especially well with low particle densities. If the particle density is high we recover less of the ground truth trajectories. In the case of constrained one-dimensional Brownian motion, where particles are trapped in nano-channels, we estimate the designated parameters well but underestimate the mis-*linking* probability. Lastly, we successfully apply our methodology to experimental data of that specific case.

When dealing with experimental data we do not cover particle disappearance or 'extra' particles due to overlapping, moving out of the focal plane, or limited fluorescing abilities. This can lead to incomplete trajectories, worse parameter estimation, and wrong calculations of mis-*linking* probabilities.

LUND UNIVERSITY

Popular Science

Simultaneous Bayesian parameter estimation and particle-tracking including calculation of mis-linking probabilities

by Lennart GOLKS

Nowadays experimentalists can visualize objects smaller than the wavelength of light, such as molecules and other nano-particles. Instead of receiving reflected light one uses directly emitted light of fluorescing molecules and can shift resolution possibilities of optical systems, which can monitor those molecules as shining dots. Working with fluorescent molecules a microscope combined with a camera can record stacks of images that can be rendered to videos. Then we can see molecules working in their environment and fulfilling their purpose which allows us to live as we do.

Following a molecule's path, we have to choose a parameter-depending model which potentially describes the motion we see under the microscope in the best way possible. Imagine being a shorter 'novisch' student going to your first party with thousand of people in a cramped room. You are being pushed and kicked by other bigger students and can barely hold your position. Now it happens to be that you usually drink a lot and you are chaotically dancing and jumping around. Next, your best friend wants to discuss something very important and is pulling you, but you do not want to leave and stop dancing, so you are drifting unsteadily and uncontrolled towards the exit. In our case, we can characterize this motion through a so-called biased Brownian motion model with two parameters: the diffusion constant and drift velocity, which are the two motion properties we like to estimate when tracking the path a particle takes.

Videos of fluorescent labeled diffusive particles are simply a lot of images in a very short time. In-between those images, the particle jumps and we only observe fluorescent dots where a particle is resting shortly. Therefore it is hard to assign consecutive positions into whole trajectories from a certain particle especially when trying to follow a lot of molecules. We have to test all possible paths we see in-between two images, as a particle could potentially take all of these steps, which is not computationally feasible. In this thesis, we tackle the problem of generating trajectories by running an assignment algorithm and acquiring a global solution, evaluated through our chosen model. Simultaneously, we estimate the drift velocity and diffusion constant. In addition, we are the first to estimate how well we can recover the particles' most likely path.

Ideally, we hope that we can set the base to find a way to quantify particle-tracking including parameter estimation. Potentially our thesis has to be extended towards more complex models to be tested and various parameters to be estimated improving the accuracy of the recovered trajectories.

Acknowledgements

Here I would like to thank all the persons who made this thesis possible, who provided very helpful tips, and who encouraged me to continue during this thesis time.

First I would like to thank my family for never stop believing in me and supporting me since day one.

For Prof. Dr. Tobias Ambjörnsson's help, I am very pleased. He guided me through this year and I benefited a lot from his comments and ideas. Overseeing the whole process including pointing out weak spots, I am very thankful to him for doing so.

When being found in the university's maze Dr. Albertas Dvirnas never hesitated to have a chat and to provide me with, sometimes well needed, hints when I got stuck.

I highly appreciate that whenever a question arose I could ask Dr. Jens Krog, Dr. Sune Levin and Dr. Jason Beech. The immediate answers could not have been any better.

I would like to thank Lauren and Connie for proofreading.

Contents

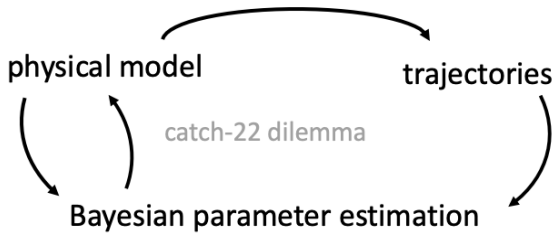
List of Figures

Chapter 1

Introduction

Super-resolution and conventional fluorescence-based microscopes produce videos showing molecules moving in their environment, experiencing certain forces, and fulfilling different purposes. We can process those videos by generating trajectories and estimate the motion parameters of the investigated particles. This process is usually divided into three different steps [1]: detection of the fluorescent dots; linking observed dots into trajectories and assigning each path to one particle; estimate the motion describing parameters.

In earlier studies, multiple algorithms were utilized to tackle the aforementioned problems in different ways. Sage et al. evaluated a wide variety of software, which localizes single molecules as white dots in realistically simulated data sets [2]. Based on this assessment the alternating descent conditional gradient (ADCG) algorithm [3] is considered to be state-of-the-art and has already been used in the Ambjörnsson group's previous work [4]. Next, algorithms linking the detected fluorescent dots into trajectories to follow the particle's path were compared by Chenouard et al [5]. They concluded that to optimize the linking process we shall know the underlying dynamics of the biological system expressed by a sophisticated physical model. However, defining the physical model by its parameters, as well as generating trajectories, requires either the not yet existing trajectories or the yet to be found parameters. This leads to a catch-22 dilemma, namely that we need the physical model and trajectories at the same time.



Challenging the mentioned catch-22 dilemma, we establish a Bayesian framework aiming to generate trajectories and estimate the physical model parameters simultaneously. Implementing a Bayesian framework provides a very quick and easy-to-implement method to optimize the physical model parameters [6]. Choosing a Brownian motion model with drift we target the best-fitting drift velocity and diffusion constant. We assess each parameter set through our Bayesian framework given the trajectories, generated by those model parameters, in one go. Here we decide to deal with the case of abstaining missing or extra dots to not over-complicate the linking process. In doing

so, we consider a so-called linear assignment problem (LAP), which we solve utilizing Munkres' algorithm also known as the Hungarian algorithm [7, 8]. The algorithm links according to the physical model and generates a solution optimizing this model globally for all obtained dots, which we detect with the ADCG algorithm. As the best-fitting model parameters are yet unknown we improve our model using a sampling approach. After sampling a new parameter set we immediately generated trajectories and assess this set through our Bayesian framework.

Once the best-fitting parameters are provided we can extend the linking algorithm with Murty's approach to be the first giving an estimate of how accurate the linking process is [9]. Murty's approach delivers ranked sets of trajectories for the best-fitting parameters. We evaluate each step in every solution with the just defined model and compare it to the first and best set of trajectories in order to provide mis-*linking* probabilities.

We first try our proposed procedure on simulated data and optimize the settings of our Bayesian framework aiming to apply it to real data. To test our methodology we simulate unconstrained two-dimensional diffusive motion of 25 particles for different box sizes. For each tested particle density, we simultaneously estimate parameters and recover trajectories. According to the most likely physical model we generate ranked solutions of the particles' paths and give mis-*linking* probabilities. If this works successfully, we simulate diffusive motion close to the experimental data we have available. Based on the results of this simulation we fine-tune our Bayesian framework. Once our proposed methodology satisfactory estimates designated parameters, recovers trajectories, and provides mis-*linking* probabilities we apply it to the experimental data. This real data contains around 240 particles trapped in nano-channels and is therefore considered to display constrained one-dimensional Brownian motion with drift.

Chapter 2

Methods

In this section, we go step by step through the methodology of this work and the physical models used in this thesis. Starting with Bayes' Theorem which will act as a basis of this work, we will guide the reader through the parameter estimation process. We will explain how we obtain positions of fluorescent dots from experimental videos followed by the linking process to recover full trajectories. Within this process, we will cover the essentials of the used algorithms, especially the nested sampling algorithm, which we will utilize to estimate designated motion parameters. Lastly, we will use Murty's approach of the assignment algorithm to estimate mis-*linking* probabilities of our proposed trajectories.

2.1 Bayesian Statistics

A Bayesian framework first mentioned more than 250 years ago provides us with a likelihood function that enables us to find the most likely parameters, given a suitable model and corresponding data [6]. The conditional probability density $p(\vec{\theta}|\text{data})$ is the posterior probability for a set of G parameters $\vec{\theta} \in [\theta_1, \ldots, \theta_g, \ldots, \theta_G]^T$. Given some observed data we construct $p(\vec{\theta}|\text{data})$ via Bayes' theorem:

$$p(\vec{\theta}|\text{data}) = \frac{p(\text{data}|\vec{\theta})P(\vec{\theta})}{P(\text{data})}, \tag{2.1}$$

where each parameter is in a range of predefined values $\theta_g \in [\theta_{g,\text{min}}\,,\, \theta_{g,\text{max}}]$ with $g = 1, \ldots, G$. In equation 2.1 the conditional probability density $p(\text{data}|\vec{\theta})$ of observed data is describing the physical model given a set of parameters $\vec{\theta}$ and is multiplied with the prior probability $P(\vec{\theta})$ of the observed parameters $\vec{\theta}$. We divide this with the probability $P(\text{data})$ of observing the data, also known as the evidence [6]. Whenever we only seek to estimate parameters $\vec{\theta}$, we can omit the evidence.

$$p(\vec{\theta}|\text{data}) \propto p(\text{data}|\vec{\theta})P(\vec{\theta}) \tag{2.2}$$

The so-called likelihood $p(\text{data}|\vec{\theta})$ enables us to evaluate a data set given the parameters $\vec{\theta}$ with a defined prior probability $P(\vec{\theta})$. The prior probability of parameters $P(\vec{\theta})$ is a uniform probability density as we herein assume not to have any pre-information on how likely the parameters are within a given range. Evaluating all parameters combinations possible would be computationally too extensive, therefore we have to sample parameters from the phase space $[\theta_{1,\min}, \theta_{1,\max}] \times \cdots \times [\theta_{G,\min}, \theta_{G,\max}]$. For further details see section 2.6.

Choosing a physical model $p(\text{data}|\vec{\theta})$ is crucial when evaluating a set of parameters through this likelihood value of some observed data. In the thesis, we want to follow diffusive particles experiencing drift. Therefore, we decide to choose a Brownian motion model characterized by a diffusion constant D and drift velocity $\vec{v} = (v_x, v_y)$. The data, with which we will feed the model, are trajectories of investigated particles that originate from a simulation first and once the proposed methodology is considered to be robust enough from an experiment. In the next section, we will introduce a d-dimensional Brownian motion model, which relates the parameters with the data.

2.2 Brownian Motion Model

To describe the motion of a diffusing particle, we use a Brownian motion model including drift velocity. We will introduce this model in d-dimensions to be able to adjust it when dealing with certain kinds of data. Then the multidimensional diffusion equation is:

$$\frac{\partial \varphi}{\partial t} = D\Big(\frac{\partial^2 \varphi}{\partial X_1^2} + \cdots + \frac{\partial^2 \varphi}{\partial X_i^2} + \cdots + \frac{\partial^2 \varphi}{\partial X_d^2}\Big), \tag{2.3}$$

which $\varphi(\Delta X_1, \ldots, \Delta X_d, \Delta t)$ has to satisfy [10]. Here $\Delta t = t_{n+1} - t_n$ refers to the time a particle needs to cover when traveling the distance $\Delta X_i = X_i^{(n+1)} - X_i^{(n)}$ in each dimension $i = 1, \ldots, d$. Assuming that one particle is starting at the origin and at the time $t_0 = 0$, the diffusion equation has the following solution:

$$\varphi(\Delta X_1, \ldots, \Delta X_d, \Delta t) = \frac{1}{(4\pi D \Delta t)^{d/2}} \exp\left[- \frac{\Delta X_1^2 + \cdots + \Delta X_d^2}{4D\Delta t} \right], \tag{2.4}$$

which is a normal distribution with the mean $\mu = 0$ and a variance $\sigma^2 = 2D\Delta t$ [11].

To include drift velocities $\vec{v} = (v_1, \ldots, v_d)$ we shift the expected value of this distribution so that our d-dimensional Brownian model with drift is:

$$\varphi(\Delta X_1, \ldots, \Delta X_d, \Delta t) = \frac{1}{(4\pi D \Delta t)^{d/2}} \exp\left[- \frac{(\Delta X_1 - v_1\Delta t)^2 + \cdots + (\Delta X_d - v_d\Delta t)^2}{4D\Delta t} \right]. \tag{2.5}$$

Before being able to evaluate paths of diffusive particles we, first, have to detect those as fluorescent dots from experimental videos, as described in section 2.4. As each of the images in an experimental video is two-dimensional, the maximum dimension of the Brownian motion d is equal to two.

2.3 Methodological Overview for Simultaneous Parameter Estimation and Linking

In this section, we would like to prepare the reader for the upcoming sections and how they interconnect with steps of the methodology.

Starting with a fluorescent video of a super-resolution microscope we extract the dots' positions out of each image in that video, see section 2.4.

Given these positions of the fluorescent dots, we generate trajectories as described in section 2.5 by using the Munkres algorithm. To assign dots in-between two images we maximize the probability of a particle taking a proposed step. The probability is defined by the Brownian motion model introduced in section 2.2, which needs the diffusion constant and drift velocity as input parameters.

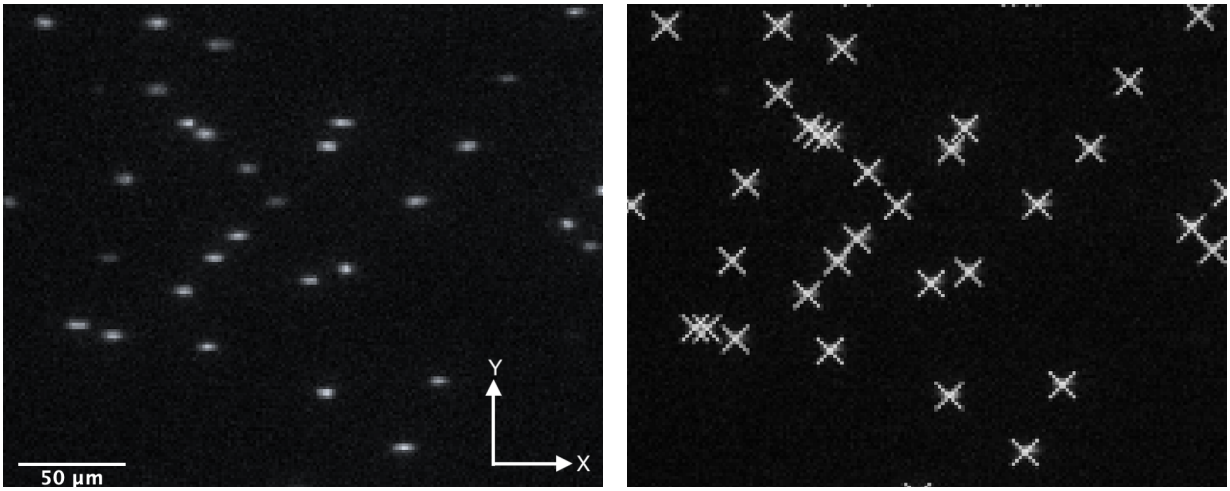
Finding those input parameters we utilize the nested sampling algorithm as described in section 2.6, where we sample a parameter set, generate trajectories, and immediately evaluate those through our Bayesian framework. If the stopping criterion of the nested sampling algorithm is met we are provided with the best-fitting parameters and the most likely trajectories.

In the section 2.7 we use those best-fitting parameters and extend the Munkres algorithm with Murty's approach. This extension gives us the possibility to calculate K ranked solutions, where the first solution is the most likely. Using these solutions we can calculate the mis-*linking* probability of each step evaluated with our Brownian motion model given the best-fitting parameters.

In the last section 2.8 of this chapter, we modify our method slightly to apply it to real data. Here we introduce nano-channels and explain how this is changing the particles' motion, as well as how we adjust the Munkres algorithm.

2.4 The Alternating Descent Conditional Gradient Method - Detection of the Dots

The output of a biophysical particle tracking experiment is a fluorescence movie, which is a stack of fluorescent images. Those fluorescence images are converted into grayscale images $I_1, \ldots, I_n, \ldots, I_N$, one of those images is seen in figure 2.1a. The time between two consecutive images is Δt. Each image is converted into a two dimensional matrix, where one pixel-point is located at (x, y) with $x \in \{1, \cdots, x_{max}\}$ and $y \in \{1, \cdots, y_{max}\}$.



(A) (B)

FIGURE 2.1: **Zoom-in of an experiment of particles in nano-channels & their corresponding detection of the ADCG algorithm**. In (A) we can see an example zoom-in of one image from experimental data generated by Sune Levin (Fredrik Westerlund's group at the Chalmers University of Technology in Gothenburg, Sweden). The white dots represent diffusive particles in nano-channels and move with a constant drift velocity in the negative x-direction. The crosses in (B) mark the observed positions generated by the alternating descent conditional gradient (ADCG) algorithm, which is explained in more detail in section 2.4.

We work with square pictures $x_{max} = y_{max}$ and the number of pixels is passed on to the algorithm beforehand.

The first step in the process of particle-tracking is the detection of the particles in each time frame. This is an inverse problem, which we solve with the alternating descent conditional gradient method (ADCG) [3]. The ADCG algorithm has proven itself to be considered state-of-the-art in solving such a problem and was used by the Ambjörnsson group in previous work [4, 2]. In our case, the inverse problem is the estimation of the number of light sources, and their respective positions, present in a given image, where one light source ideally represents one molecule. In practice, we use a modified *Julia* code of Schiebinger et al., which implements all the steps described below to solve this inverse problem of detecting the dots [3].

To detect all molecules captured in one image the ADCG algorithm follows a set procedure. In the ADCG method, first, we assume that each illumination source emits light like a Gaussian-shaped point spread function (PSF), and second, a combination of such sources is creating the analyzed image. Depending on the emitted wavelength of the fluorescent molecules and the numerical aperture of the used camera system, we define the PSF accordingly. We divide the image into a grid such that the amount of pixels is lower than the number of grid points, aiming to increase accuracy. Once we

remove a set noise from the image, we let the algorithm run a predefined number of iterations, where it creates or removes PSF on a constructed grid. The goal is to find optimal positions and intensities of the PSFs in this artificial picture so that it matches the original picture in the best way possible. Therefore a loss function evaluates the difference between the image, which is generated by the algorithm, and the original image. Based on the intensity difference of each pixel weighted with this loss function and a chosen factor the images are compared and the algorithm stops once the images are similar enough, for more details we refer to Dolev Illouz's bachelor thesis [4]. Finally, the algorithm provides the required positions and intensities of the light sources present in the analyzed image, as seen in figure 2.1b [3].

We proceed through every given image I_n and observe positions $\{(x_n^{(f)}, y_n^{(f)})\}$ of the dots $f = 1, \ldots, F$ present as light sources in the images. The next non-trivial step is to link these observed dots in-between two consecutive images.

2.5 Assigning the Dots

Finding the best assignment of the observed dots according to the model is an optimization problem that we solve using an algorithm based on the Hungarian algorithm [8]. To link observed dots between two consecutive time frames we set up a cost matrix $C = (P \times P)$, where we enumerate according to the number of dots $1, \ldots, p, \ldots, P$ present in a picture. The goal is to find a linking of observed dots in-between time frames n and $n+1$ such that the posterior probability is maximized according to the predefined Brownian motion model. We achieve this by minimizing the negative logarithm of the posterior probability of our model given by Eq. 2.5. The probability for linking two dots does depend on the drift velocity and on the diffusion constant, which we both consider being the same for all dots.

This leads to the entries of the cost matrix for a combination of two positions of the dots p and p' at step n and the following step $n+1$, with preset values for the drift velocities $(v_1, \ldots, v_d)$ in d dimensions and the diffusion constant D:

$$C_{p,p'} = \frac{\left(x_{(1,n)}^{(p)} - x_{(1,n-1)}^{(p')} - v_1 \Delta t\right)^2 + \cdots + \left(x_{(d,n)}^{(p)} - x_{(d,n-1)}^{(p')} - v_d \Delta t\right)^2}{4D\Delta t} + \frac{d}{2} \ln\left[4\pi D \Delta t\right] . \tag{2.6}$$

The maximum number of dimension we have to consider in our case is two, but we adjust our cost matrix entries depending on the data we are treating. For simplicity reasons we will continue with one explicit case, which is a two-dimensional Brownian motion model including drift velocities in the x- and y-direction. This leads to the cost matrix entries:

$$C_{p,p'} = \frac{\left(x_n^{(p)} - x_{n-1}^{(p')} - v_x \Delta t\right)^2 + \left(y_n^{(p)} - y_{n-1}^{(p')} - v_y \Delta t\right)^2}{4D\Delta t} + \ln\left[4\pi D \Delta t\right] . \tag{2.7}$$

Here we can already point out that the additive factor $\ln[4\pi D\Delta t]$ and scaling factor $1/(4D\Delta t)$ does not influence the assignment. As a consequence, all values of D in combination with one drift velocity will have the same best linking.

Solving this assignment problem we use Munkres' algorithm, which is a variation of the so-called Hungarian algorithm [7]. The algorithm provides us with the best assignment between two consecutive time frames. Doing this for every two consecutive time frames we can link together whole trajectories of particles $S_{\text{best}} = \{(X_n^{(p)}, Y_n^{(p)})\}$, where $p = 1, \ldots, P$ and $n = 1, \ldots, N$, which is the best assignment we can compute. The maximum number of needed operations to solve this assignment problem is in the order of $\mathcal{O}((N-1)P^3)$, and lies within the range of the most efficient algorithms [7]. The linking process with defined in- and outputs is summarized in algorithm 1.

In the following chapter, we evaluate the best set of trajectories S_{best} with a likelihood value based on our Brownian motion model to estimate the diffusion constant and drift velocities.

Algorithm 1: Munkres' Algorithm

1 **Input**: Dot positions $\{(x_n^{(f)}, y_n^{(f)})\}$, where $f = 1, \ldots, F$ and $n = 1, \ldots, N$, drift velocity $\vec{v} = (v_x, v_y)$;
2 **Setup**: Cost matrix $C = (P \times P)$ according to Eq. 2.7;
3 **Output**: Best set of trajectories S_{best}

2.6 Parameter Estimation

To find the best-fitting parameters $\theta_1^*, \ldots, \theta_G^*$ we follow a set procedure and have to sample a set of parameters in the phase space $[\theta_{1,min}, \theta_{1,max}] \times \cdots \times [\theta_{G,min}, \theta_{G,max}]$ as it is computationally too expensive to calculate all possible combinations.

We evaluate a chosen set of parameters $\vec{\theta} = (\theta_1, \theta_2, \theta_3)$ when calculating the posterior probability value $p(\theta_1, \ldots, \theta_G | S_{\text{best}})$ for a given set of trajectories S_{best} as seen in figure 2.2. First, we implement a model with parameters, which is in our case a Brownian motion model with drift as described in section 2.2. Based on this model we are assigning the fluorescent dots in-between two images, which are obtained in section 2.4 given a set of parameters. In a two-dimensional case the physical model parameters are $(\theta_1, \theta_2, \theta_3) = (D, v_x, v_y)$, where D is the diffusion constant and (v_x, v_y) are the drift velocities in x- and y-direction. According to a cost function defined by our model, we maximize the probability of particles moving in-between two images and set up a cost matrix, as seen in Eq. 2.7. Using Munkres' algorithm as introduced in section 2.5 we get a set of best trajectories S_{best}, which is independent to D. To find the best-fitting diffusion constant D, we calculate the posterior probability $p(D, v_x, v_y | S_{\text{best}})$ for a diffusion constant in the range of $[D_{\text{min}}, D_{\text{max}}]$ and chosen drift velocities, as seen in the equation below:

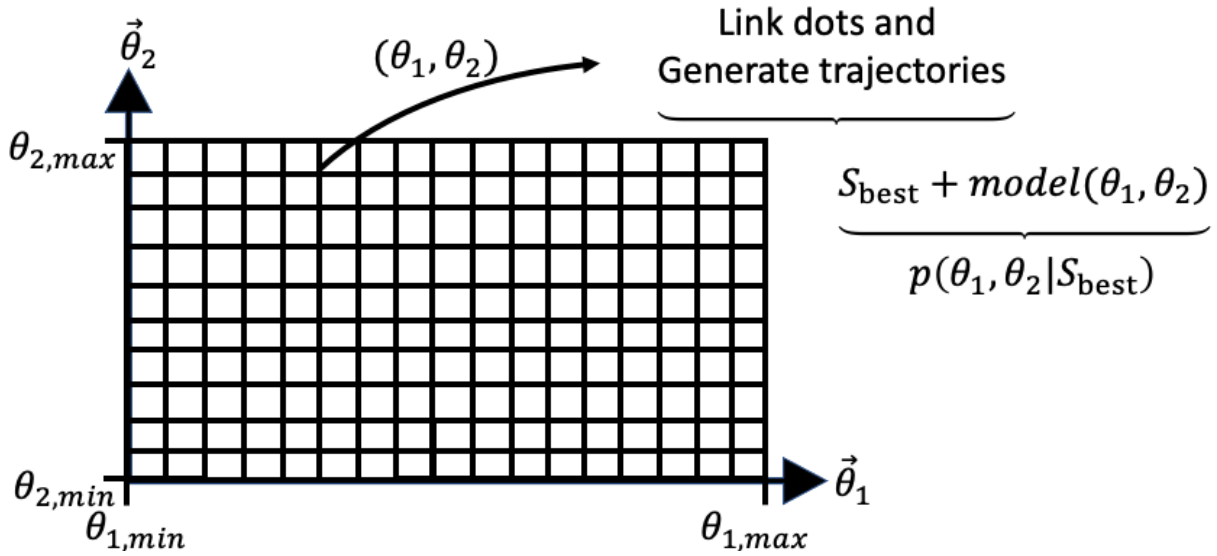


FIGURE 2.2: **Schematic illustration of parameter evaluation in a two-dimensional case.** Given two parameters (θ_1, θ_2) picked out of the parameter space $[\theta_{1,min}, \theta_{1,max}] \times [\theta_{2,min}, \theta_{2,max}]$ and observed data, which in our case is the observed positions of the fluorescent dots, we can set up a cost matrix as described in section 2.5. Using the assignment algorithm 1 we can link the positions of the observed dots and generate a set of best trajectories S_{best}. Using a physical model, e.g. our Brownian motion model with drift, we can calculate the posterior distribution $p(\theta_1, \theta_2 | S_{\text{best}})$ as seen in Eq. 2.8. This is a simplified description of how we evaluate a set of parameters, in section 2.6 we explain the process of how we sample the most likely parameters in the parameter space.

$$p\Big(D, \vec{v} | \big(X_1^{(1)}, X_1^{(1)}\big), \ldots, \big(X_n^{(p)}, Y_n^{(p)}\big), \ldots, \big(X_N^{(P)}, Y_N^{(P)}\big)\Big) \propto \prod_{n=2}^{N} \prod_{p=1}^{P} \frac{1}{4D\pi\Delta t} \exp\left[- \frac{\big(X_n^{(p)} - X_{n-1}^{(p)} - v_x \Delta t\big)^2 + \big(Y_n^{(p)} - Y_{n-1}^{(p)} - v_y \Delta t\big)^2}{4D\Delta t} \right]. \tag{2.8}$$

Equation 2.8 describes the posterior probability for P particles taking $N-1$ steps within a time $(N-1)\Delta t$ in a proposed linking $S_{\text{best}} = \{\big(X_n^{(p)}, Y_n^{(p)}\big)\}$ with $p = 1, \ldots, P$ and $n = 1, \ldots, N$ for a specific set of parameters. The maximum posterior probability $\max_{D \in [D_{\min}, D_{\max}]} p(D, \vec{v} | S_{\text{best}})$ of the best solution S_{best} is our benchmark, which we compare to other sets of parameters.

Dealing with the complexity of estimating the drift velocity and diffusion constant, we use the nested sampling approach similar to *Bayesian model selection with fractional*

Brownian motion by Krog et al. [12]. Usually the algorithm is used to calculate the evidence $P(\text{data})$ (see Eq. 2.1) by randomly sampling over the parameter space. In our case, we use an 'optional by-product' of the nested sampling algorithm, which is finding the best drift velocity $\vec{v}^* = (v_x^*, v_y^*)$ and best diffusion constant D^* for a given set of detected dots [13]. The nested sampling algorithm is a Monte Carlo method and follows a set procedure until a stopping criterion is reached. We start by randomly sampling U drift velocities $\vec{v}_1, \ldots, \vec{v}_u, \ldots, \vec{v}_U$ within the range $[v_{x,\min}, v_{x,\min}] \times [v_{y,\min}, v_{y,\max}]$. For each of those samples we assign (independently of D) the fluorescent dots, observed in section 2.4, to a set of best trajectories $S_{\text{best},u}$. We then compute the corresponding posterior probability $p\left(D, \vec{v}_u | S_{\text{best},u}\right)$, which we maximize with respect to D in the preset range of $[D_{\min}, D_{\max}]$. Within this process we find the optimal diffusion constant D_u expressed through a maximal posterior value $\max_{D\in[D_{\min},D_{\max}]} p\left(D, \vec{v}_u | S_{\text{best},u}\right)$, which we assign to the sample $\left(D_u, \vec{v}_u\right)$.

To find the overall best-fitting parameters $\left(D^*, \vec{v}^*\right)$, we update our U parameter samples. In doing so we remove the least probable parameter combination $\min_{D_u\in[D_1,\ldots,D_U],\vec{v}_u\in[\vec{v}_1,\ldots,\vec{v}_U]} p\left(D_u, \vec{v}_u | S_{\text{best},u}\right)$ and replace it with a more likely sample $\left(D', \vec{v}'\right)$. After calculating the best linking S'_{best} for the proposed $\vec{v}'$ and a posterior probability $p\left(D, \vec{v}' | S'_{\text{best}}\right)$, we optimize this posterior value to find the best diffusion constant D' at $\max_{D\in[D_{\min},D_{\max}]} p\left(D, \vec{v}' | S'_{\text{best}}\right)$. If a new proposed candidate $(D', \vec{v}')$ is less likely than the one we want to remove, we reject this try and sample again until we find a more likely candidate or the number of rejected tries n_{tries} is equal to a preset number of allowed tries N_{sweep}. According to the fraction of rejected moves within the number of tries $R = n_{\text{tries}}/N_{\text{sweep}}$ we reduce the parameter space equally for each parameter if needed, aiming to sample from a region in which it is very likely to find the best drift velocity. Else we add $\left(D', \vec{v}'\right)$ to our sampling space assigned with $p\left(D', \vec{v}' | S'_{\text{best}}\right)$. The removed sample is added with a weight factor to the current estimate of the evidence and the remaining evidence is calculated, as introduced by Skilling et al. and executed by Krog et al. [13, 12]. While the fraction of the remaining evidence divided by the current estimate of the evidence is greater than a threshold we repeat the procedure. Once the stopping criterion is reached the algorithm will provide us with the overall best-fitting parameter combination $\left(D^*, \vec{v}^*\right) = \max_{D_u\in[D_1,\ldots,D_U],\vec{v}_u\in[\vec{v}_1,\ldots,\vec{v}_U]} p\left(D_u, \vec{v}_u | S_{\text{best},u}\right)$ and the corresponding best solution S_{best}.

We summarized our nested sampling approach in the algorithm 2 below, which will enable us to continue with the calculation of the mis-*linking* probabilities for the investigated particles given the best-fitting parameters.

2.7 Mis-*linking* Probabilities

Given the results of the nested sampling algorithm in section 2.7 we can estimate mis-*linking* probabilities by reusing Munkres' algorithm. Murty's variant of Munkres' algorithm from 1968 is assigning the in section 2.4 acquired fluorescent dots based on the

Algorithm 2: Nested Sampling

Input: Sample size U, range of diffusion constants $\vec{D} = [D_{\min}, D_{\max}]$, range of drift velocities $\vec{v}_x = [\vec{v}_{x,\min}\vec{v}_{x,\min}]$ and $\vec{v}_y = [\vec{v}_{y,\min}, \vec{v}_{y,\max}]$, dots position $\{(x_n^{(f)}, y_n^{(f)})\}$ where $f = 1, \ldots, F$ and $n = 1, \ldots, N$, and stopping criterion ϵ;
Sample: Sample $\vec{v}_1, \ldots, \vec{v}_u, \ldots, \vec{v}_U$ from a uniform distribution within the range $[\vec{v}_{x,\min}\vec{v}_{x,\min}] \times [\vec{v}_{y,\min}, \vec{v}_{y,\max}]$;
for $u = 1$ **to** U **do**
    Calculate: $S_{\text{best},u}$ use algorithm 1;
    Compute: posterior probability distribution $p(D, \vec{v}_u | S_{\text{best},u})$;
    Find: $(D_u, \vec{v}_u)$ at $\max_{D \in \vec{D}} p(D, \vec{v}_u | S_{\text{best},u})$;
end
Calculate: Remaining evidence and current estimate of evidence;
while $\frac{\textit{remaining evidence}}{\textit{current estimate of evidence}} > \epsilon$ **do**
    Propose: $\vec{v}'$ within parameter space;
    Calculate: S'_{best} and $p(D, \vec{v}' | S'_{\text{best}})$;
    Find: $(D', \vec{v}')$ at $\max_{D \in \vec{D}} p(D, \vec{v}' | S'_{\text{best}})$;
    Set: $n_{\text{try}} = 0$;
    while $n_{try} \leq N_{Sweep}$ **and** $p(D', \vec{v}' | S'_{best}) < \min_{D_u \in [D_1,\ldots,D_U], \vec{v}_u \in [\vec{v}_1,\ldots,\vec{v}_U]} p(D_u, \vec{v}_u | S_{best,u})$ **do**
        Set: $n_{\text{try}} = n_{\text{try}} + 1$;
        Propose new: $\vec{v}'$ within parameter space;
        Calculate: S'_{best} and $p(D, \vec{v}' | S'_{\text{best}})$;
        Find: $(D', \vec{v}')$ at $\max_{D \in \vec{D}} p(D, \vec{v}' | S'_{\text{best}})$;
    end
    if $p(D', \vec{v}' | S'_{best}) > \min_{D_u \in [D_1,\ldots,D_U], \vec{v}_u \in [\vec{v}_1,\ldots,\vec{v}_U]} p(D_u, \vec{v}_u | S_{best,u})$ **then**
        Update: $\min_{D_u \in [D_1,\ldots,D_U], \vec{v}_u \in [\vec{v}_1,\ldots,\vec{v}_U]} p(D_u, \vec{v}_u | S_{\text{best},u}) = p(D', \vec{v}' | S'_{\text{best}})$;
    end
    if $n_{\text{try}} = N_{\text{Sweep}}$ **then constrain**: Parameter space $P(\vec{v}_x, \vec{v}_y)$, see Eq. 3.1;
    Calculate: Remaining evidence and current estimate of evidence;
end
Output: most likely drift velocity $\vec{v}^* = (v_x^*, v_y^*)$ and diffusion constant D^* at $\max_{D_u \in [D_1,\ldots,D_U], \vec{v}_u \in [\vec{v}_1,\ldots,\vec{v}_U]} p(D_u, \vec{v}_u | S_{\text{best},u})$ with a best proposed solution S_{best}

cost introduced through Eq. 2.7. In addition, this approach provides us with K solutions $S_1, \ldots, S_K$ ranked in 'increasing order of cost', so that $S_1 = S_{\text{best}}$ [9].

Next, we express the *linking* probability of a particle taking one step in-between two images I_n and I_{n+1} in the k^{th} best solution given the set of best-fitting parameters $\left(D^*, \vec{v}^*\right)$. Equation 2.9 gives us this probability for a two dimensional movement of a particle p making a proposed move according to the Brownian motion model with drift defined in section 2.2:

$$\frac{1}{4D^*\pi\Delta t} \exp\left[- \frac{\left(X_n^{(p,k)} - X_{n-1}^{(p,k)} - v_x^*\Delta t\right)^2 + \left(Y_n^{(p,k)} - Y_{n-1}^{(p,k)} - v_y^*\Delta t\right)^2}{4D^*\Delta t} \right]. \tag{2.9}$$

To give an estimate of how well we recover trajectories we compare the best solution S_{best} to the other K solutions we acquired using Murty's approach of the linking algorithm. In-between two consecutive images I_n and I_{n+1} we pick one step of a particle p in the best solution $S_1 = S_{\text{best}}$ given the parameters $\left(D^*, \vec{v}^*\right)$ with the starting position $\left(X_n^{(p,1)}, Y_n^{(p,1)}\right)$ and ending position $\left(X_{n+1}^{(p,1)}, Y_{n+1}^{(p,1)}\right)$. If a particle p' in the k^{th} solution is starting and ending at the same positions as the particle p in the best solution, we add the probability of taking that step expressed through Eq. 2.9, to the nominator of our *linking* probability in Eq. 2.10. The denominator of Eq. 2.10 is the sum of all probabilities calculated with Eq. 2.9, where a particle is taking a step starting at $\left(X_n^{(p,1)}, Y_n^{(p,1)}\right)$ within two consecutive images I_n and I_{n+1} in every obtained solution for the most likely drift velocity. The mis-*linking* probability is equal to 1−equation 2.10.

$$\frac{\text{sum of Eq. 2.9 for all particles matching } S_1 \text{ within } I_n \text{ and } I_{n+1} \text{ in all } S_k}{\text{sum of Eq. 2.9 for all particles within } I_n \text{ and } I_{n+1} \text{ in all } S_k} \tag{2.10}$$

To find a well-fitting K, which is the number of solutions that we calculate using Murty's approach, is crucial in order to estimate a realistic mis-*linking* probability. If we calculate a mis-*linking* using too small a value of K we would obtain unrealistic results by not considering other possible solutions. Therefore, we increase K and recover one additional solution as long as the mis-*linking* probability is changing. After each increment of K, we calculate the mis-*linking* probability with the best-fitting parameters $\left(D^*, \vec{v}^*\right)$ given by the nested sampling algorithm. Once we reach a stable average mis-*linking* probability of overall steps in one data set we stop increasing K and obtain the final mis-*linking* probabilities for each step a particle is taking.

Below you find an overview of the mis-*linking* calculations including the procedure of finding an optimal number of calculated solutions.

Algorithm 3: Calculation of Mis-*linking* Probabilities

Input: Best sampled parameters $\left(D^*, \vec{v}^*\right)$ from algorithm 2, observed dots $\{\left(x_n^{(f)}, y_n^{(f)}\right)\}$, where $f = 1, \ldots, F$ and $n = 1, \ldots, N$;
Set: $K = 1$;
Set: average mis-*linking* probability $= 0$;
while mis-*linking* probability is increasing do
    Increase: $K = K + 1$;
    Calculate: $S_1, \cdots, S_K$ given $\left(D^*, \vec{v}^*\right)$ use algorithm 1;
    Take: best trajectories S_1;
    for $k = 1$ **to** K **and;** $p = 1$ **to** P **and;** $n = 1$ **to** N **do**
        if Step in S_1 = Step in S_k **then**
                Add: *linking*-probability (Eq. 2.9) to nominator of 2.10;
                Add: *linking*-probability (Eq. 2.9) to denominator of;
            else Add: *linking*-probability (Eq. 2.9) to denominator of Eq. 2.10;
        end
    end
    Calculate: average mis-*linking* probability;
end
Output: mis-*linking*-probability = 1-*linking*-probability for particles taking steps in the best solution S_1 given the best-fitting parameters $\left(D^*, \vec{v}^*\right)$ and number of calculated solutions K

2.8 Generate Trajectories and Estimate Parameters for Experimental Data

When dealing with experimental data we slightly adjust our approach. The experimental data we have available shows particles in nano-channels as white dots. These nano-channels do constrain the motion of the particles in the y-direction. Consequently, the particles diffuse only with a drift velocity in the x-direction.

The reason for the adjustment of the linking algorithm is that dot positions may be missing due to experimental noise and imperfections in the data. The limiting fluorescence abilities of a molecule lead to on and off switching of the fluorescent label attached to it. In combination with a finite lifetime, we are likely to miss out on white dots within a biological fluorescent-based experiment. Additionally, molecules can overlap or move out of the focal plane. In all of these cases, the ADCG algorithm, which is detecting the fluorescent dots, will most likely not recognize those molecules. This certainly leads to "extra" dots but also missing dots in the images of the experimental data. Unlike in simulations, where we assume no disappearing or additional particles, we have to filter out those positions and leave them unassigned.

Murty's algorithm as implemented in *MATLAB* gives us the opportunity to define a threshold, which leaves positions unassigned. This is important because if we do assign positions with a high cost, we obtain very unlikely and unrealistic jumps of particles that are magnitudes higher compared to most other steps in the investigated data set. This threshold is coupled to the costs of assignments and defines the maximum cost of a proposed step within the linking process. Positions are unassigned if the cost of an assignment is too 'expensive' and over-exceeds the cost of non-assignment. Our choice of cost is defined in Eq. 2.7 and dependent on the step size a particle takes including the shift caused by the drift velocity. The diffusion constant acts as a scaling factor and will not change the assignment. We reduce this cost of non-assignment and recursively generate trajectories until the largest step a particle takes in one data set is not greater than the median plus two times the interquartile range (IQR) of all the step sizes in one data set.

Another fact we can make use of is that the particles are trapped in nano-channels. By first identifying the nano-channels and only considering the particles inside each nano-channel we can increase computational speed as the cost matrices are considerably smaller and easier to solve for the assignment algorithm. Localizing and estimating the width of the nano-channels in the experimental data we use a function written by Sune Levin (Fredrik Westerlund's group at the Chalmers University of Technology in Gothenburg, Sweden). In the case of simulations, we know where the nano-channels are as well as their width and pass that on the Munkres algorithm. We do not permit particles to jump in-between different nano-channels and therefore follow a one-dimensional diffusive motion with drift. Consequently, we adjust our physical model

to a one-dimensional Brownian motion model with drift:

$$\varphi(\Delta x, \Delta t) = \frac{1}{\sqrt{4\pi D \Delta t}} \exp\left[-\frac{(\Delta x - v_x \Delta t)^2}{4 D \Delta t} \right]. \tag{2.11}$$

Aiming to estimate the diffusion constant D and the drift velocity in the x-direction v_x we use our Bayesian framework to simultaneously generate trajectories of the molecules appearing as white dots.

We give an overview of this refined linking process in the box below, where in addition to assigned fluorescent dots we obtain unassigned dots which partly leads to incomplete trajectories. We use those trajectories to recursively estimate the designated parameters using our Bayesian framework.

Algorithm 4: Munkres' Algorithm for Experimental Data

Input: Positions $\left(x_n^{(1)}, y_n^{(1)}\right), \ldots, \left(x_N^{(F)}, y_N^{(F)}\right)$, nano-channel locations, nano-channel width, drift velocity $\vec{v} = (v_x, v_y)$;
while Median(step sizes) + 2 IQR(step sizes) < max(step sizes) do
    Reduce: Cost of non-assignment;
    for each nano-channel do
        Find: all P particles inside that nano-channel;
        Setup: Cost matrix $C = (P \times P)$ according to Eq. 2.7;
        Calculate: Best set of trajectories S_{best};
    end
    Calculate: Step sizes for all particles and all steps;
end
Output: Best set of trajectories S_{best}

Chapter 3

Results

In this chapter, we present the simulation setup which we use to obtain the following results of the methodology introduced in chapter 2. By guiding the reader towards the experimental outcomes we will go through the most important results to understand the process of calculating mis-*linking* probabilities and estimate parameters besides generating trajectories simultaneously.

3.1 Setup of the Simulation and Bayesian Framework

In this section, we go through the simulation setup and the setup of the Bayesian framework we use to estimate the parameters of the physical model, which we summarize in table 3.1. We simulate diffusive motion for different particle densities and do the analysis in *MATLAB*.

Based on the Smoluchowski equation we generate diffusive motion and follow a 'basic stochastic simulation algorithm' (SSA) introduced on pg. 17 in [11]. To include a drift velocity we shift each simulated diffusive step with $\mu = \vec{v}\Delta t$. Within our simulations, we have no unassigned or overlapped positions of any particles. We simulated round particles with a radius of $R = 45 - 50$nm in water with a viscosity of $\eta \approx 1\text{mPa} \cdot \text{s}$ at room temperature $T = 293$K, which gives a diffusion constant of $D \approx k_B T/6\pi\eta R = 4\mu\text{m}^2/\text{s}$, where k_B is the Boltzmann constant [10]. In doing so we keep a constant number of diffusive particles in a quadratic box with periodic boundaries, which has a different length in each simulation scaled with the number of pixels. After simulating the trajectories we randomize the positions in each time frame to make sure that no pre-information is transmitted when linking the dots with Munkres' algorithm.

When constraining the diffusive motion to simulate data close to the real setup we create nano-channels that are 300nm wide as it is the case in the experimental data.

To estimate the diffusion constant and drift velocity we utilize a Bayesian framework and the nested sampling algorithm aiming to maximize the posterior probability at $p\left(D^*, \vec{v}^* | S_{\text{best}}\right)$. The posterior distribution $p\left(D^*, \vec{v}^* | S_{\text{best}}\right)$ in Eq. 2.8 is not normalized, therefore, we deal with very small numbers when calculating probability values for longer trajectories and many particles. To avoid this, we convert the maximization

problem into a minimization problem and work with the negative logarithmic function of the posterior probability $-\ln[p(D^*, \vec{v}^*|S_{\text{best}})]$.

To find the best-fitting drift velocity and diffusion constant we use the nested sampling algorithm implemented as mentioned in section 2.7, where we take 100 samples in total from 1000 possible parameter values in each dimension. The range in which we start sampling drift velocities and diffusion constants is defined in table 3.1. Within this algorithm, we set the number of allowed tries to find a more likely sample to $N_{\text{sweep}} = 30$. We constrain the sampling space of drift velocities l_u, centered on the most likely drift velocity tuple, when the fraction of rejected moves R within N_{sweep} is greater than 50%:

$$l_u \to \min(l_u \exp(0.5 - R), 1). \tag{3.1}$$

The algorithm is terminated once the difference between the remaining and current evidence is smaller than $\epsilon = 10^{-5}$.

Parameter	Numerical Value	Unit	Description
D_{sim}	4	$\mu\text{m}^2/\text{s}$	Simulated diffusion constant
$\min \vec{D}$	0.05	$\mu\text{m}^2/\text{s}$	Lower boundary for diffusion constant
$\max \vec{D}$	500	$\mu\text{m}^2/\text{s}$	Upper boundary for diffusion constant
$v_{x,\text{sim}}$	-55	$\mu\text{m}/\text{s}$	Simulated drift velocity in the x-direction
$v_{y,\text{sim}}$	0	$\mu\text{m}/\text{s}$	Simulated drift velocity in the y-direction
$\min \vec{v}_x$	-0.1	mm/s	Lower boundary for v_x
$\max \vec{v}_x$	0	mm/s	Upper boundary for v_x
$\min \vec{v}_y$	-0.05	mm/s	Lower boundary for v_y
$\max \vec{v}_y$	0.05	mm/s	Upper boundary for v_y
Δt	0.035	s	Waiting time in-between steps
Pixel length	0.431	μm	Length of a quadratic pixel
P	25 or 240		Number of particles
N-1	5		Number of steps
U	100		Sample size
ϵ	10^{-5}		Nested sampling stopping criterion

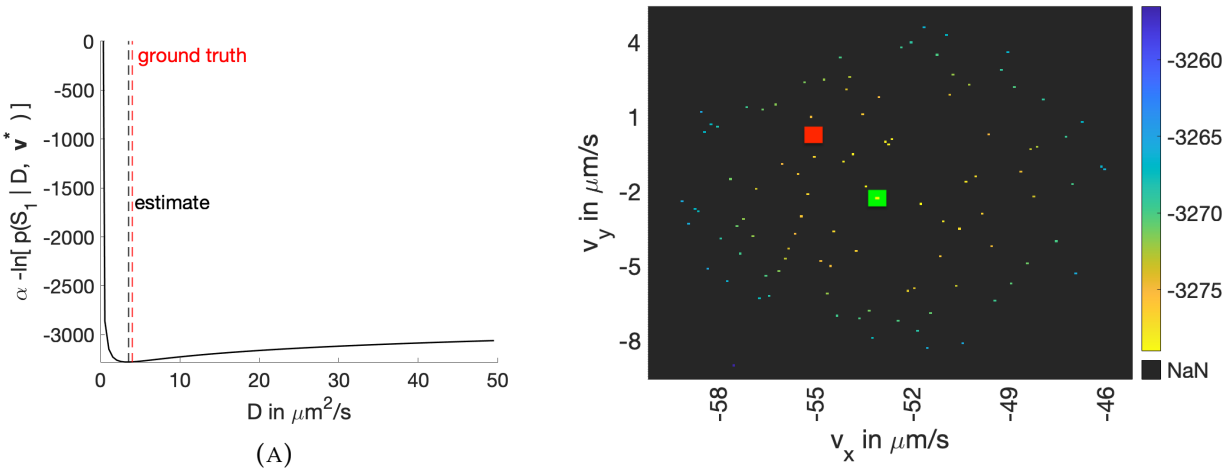
TABLE 3.1: Fixed numerical values used in the simulation and evaluation of this study.

We analytically estimate the error σ_{θ^*} of the parameters by calculating the absolute second derivative of the logarithmic posterior probability evaluated at most likely parameter θ^*:

$$\sigma_{\theta^*} = \left(- \frac{d^2 \ln \left[p(\theta|S_{\text{best}}) \right]}{d\theta^2} \bigg|_{\theta^*} \right)^{-1/2} \text{[6]}. \tag{3.2}$$

3.2 Unconstrained Diffusive Motion

Here we present the estimation of the diffusion constant and drift velocity optimizing our physical model in Eq. 2.5, given the most likely recovered trajectories and its mis-*linking* probabilities. In this section, we consider unconstrained Brownian motion with drift of 25 particles in two dimensions, for further details we refer to section 3.1 and table 3.1.



(A) (B)

FIGURE 3.1: **Finding the best-fitting parameters if the stopping condition of the nested sampling algorithm is met.** In figure (A) we plot a function proportional to the negative logarithm of the posterior probability $-\ln[p(D,\vec{v}^*|S_{\text{best}})]$ for the most likely drift velocity $\vec{v}^*$ obtained in figure (B), given a set of best-fitting trajectories S_{best}. The most likely diffusion constant $D^* = 3.6 \pm 0.7\mu\text{m}^2/\text{s}$ lies at the minimum of that function. Each of the data points in figure (B) represents a drift velocity sample $\vec{v}_u$ and its minimal posterior probability $-\ln[p(D_u,\vec{v}_u|S_{\text{best}})]$ at the most likely diffusion constant D_u, indicated by the color bar. If the stopping criterion of the nested sampling approach is met we can specify the most likely drift velocity $\vec{v}^* = (-53,2) \pm (2,2)\mu\text{m}/\text{s}$ at the overall minimum, which is marked in green. The nested sampling approach reduces the parameter space from which we sample and shows that the samples lie within a region in which we are likely to find a parameter combination close to the ground truth, which is marked in red. The estimation fits well with the red marked simulation input of $D_{\text{sim}} = 4\mu\text{m}^2/\text{s}$ and $\vec{v}_{\text{sim}} = (-55,0)\mu\text{m}/\text{s}$. The simulation was setup according to section 3.1 in a quadratic box with a size of 200×200 pixels and periodic boundaries.

In figure 3.1 (A) we can see a function proportional to the negative logarithmic posterior probability of the best-fitting parameter set $-\ln[p(D,\vec{v}^*|S_{\text{best}})]$, given the corresponding best set of trajectories S_{best}. The posterior probability function seen in figure

3.1 (A) for the best-fitting drift velocity $\vec{v}^*$ has a minimum at the best-fitting diffusion constant $D^* = 3.6\mu\text{m}^2/\text{s}$. This is a close estimate to the simulation input marked in red.

The most likely drift velocity of $\vec{v}^* = (-53, 2) \pm (2, 2)\mu\text{m/s}$ lies at the minimum of the sampled data points in figure 3.1 (B) and is marked in green. We marked the ground truth parameters in red, as seen in figures 3.1 (A) and 3.1 (B). Figure 3.1 (B) shows the stopping criterion of the nested sampling algorithm, where each data point represents the minimum of the negative logarithmic posterior function of a specific drift velocity and diffusion constant, given the best set of trajectories S_{best}. These trajectories and posterior probabilities are simultaneously generated for each sampled parameter set. The estimation is close to the simulation input and shows that the Bayesian framework based on our Brownian motion model with drift works for the tested simulations. The specific simulation setup to obtain this graph is a box with the size of 200×200 pixels.

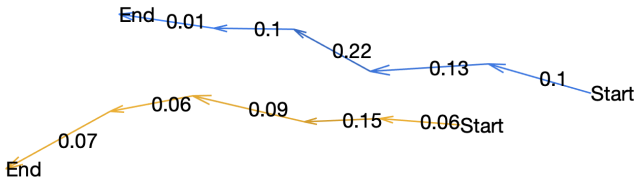


FIGURE 3.2: **Zoom-in of particles' most likely trajectory with calculated mis-*linking* probabilities.** We link positions of dots according to Munkres' algorithm based on the model in Eq. 2.5. Following the arrows, we display the particles' most likely path, with a mis-*linking* probability indicated by the numbers. Calculating the three best sets of trajectories using the most likely parameter values, as described in figure 3.1, we can estimate the mis-*linking* probability as introduced in section 2.7. The 25 particles diffuse unconstrained in a box with a size of 200×200 pixels. The simulation input values are described in table 3.1.

Given the most likely parameters $(D^*, \vec{v}^*)$, provided by the nested sampling algorithm in figure 3.1, we can calculate the mis-*linking* probabilities for each step, as seen in figure 3.2. This figure shows a zoom-in of two trajectories from the best set of trajectories. Here we can see two very close particles taking five steps and following the arrows in-between the 'Start' and the 'End' sign. The calculated mis-*linking* probabilities are attached to the arrows and are based on the three best sets of trajectories. If the mis-*linking* probabilities are higher we are less likely to have recovered the ground truth.

After estimating the diffusion constant and drift velocity we can obtain the overall mis-*linking* probability in one simulation when averaging overall calculated mis-*linking* probabilities of every step. In doing so, we assess our methodology for different particle densities with a changing box size and constant number of particles.

To test the behavior of our proposed methodology we simulated diffusive motion for different particle densities. Low particle densities include trajectories, which are

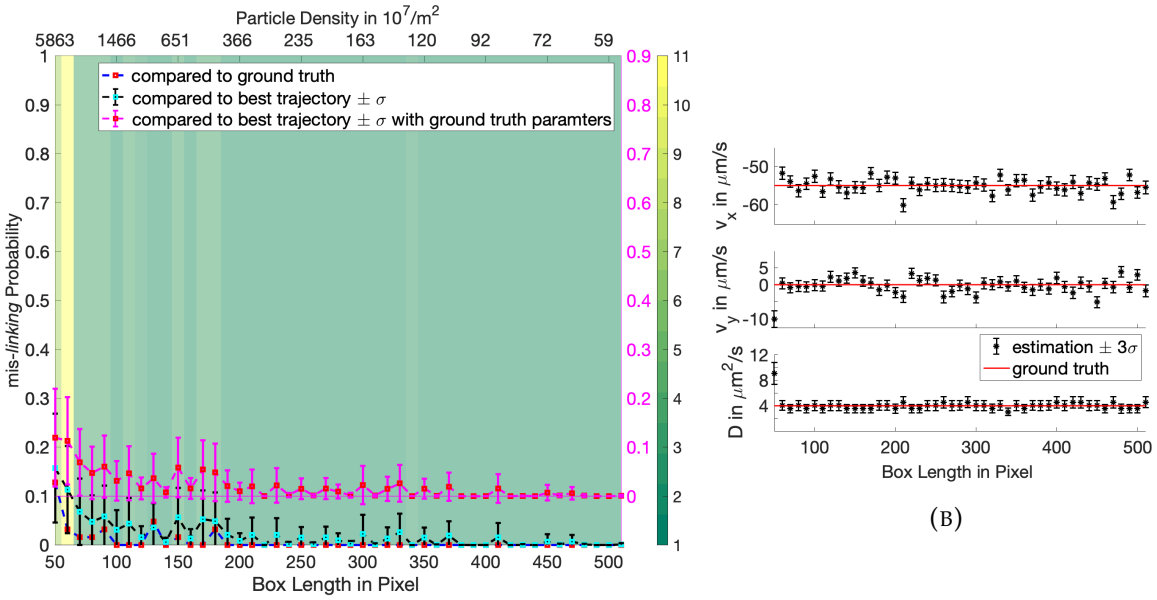


FIGURE 3.3: **Mis-*linking* estimates compared to ground truth values & estimated model parameters**. The results shown here were obtained by simulated data to confirm our methodology. We simulate 25 particles taking five unconstrained steps in a quadratic box with periodic boundaries for changing box sizes. We summarize all important parameters for the simulation and estimation process in table 3.1. Figure (A) shows the average mis-*linking* probability in a data set, which is calculated as introduced in chapter 2.1. The calculated mis-*linking* probability using the simulation input parameters $(D_{\text{sim}}, \vec{v}_{\text{sim}})$ in magenta has an offset of 0.1 as it is very similar to the mis-*linking* probability calculated with the best estimate $(D^*, \vec{v}^*)$, marked in black with green squared markers. The calculated mis-*linking* probabilities slightly overestimate the actual mis-*linking* probabilities (blue graph with red squares), which we obtain by comparison of the best set of recovered trajectories $S_{\text{best}} = S_1$ to the simulated trajectories for each box size. The background color refers to the number of calculated solutions K using Murty's approach of the Munkres' algorithm, which is indicated by the color bar and lies within the range of three solutions for lower particle densities and up to eleven calculated solutions for higher particle densities. We can see that we need to calculate more solutions for higher mis-*linking* probabilities and higher particle densities to obtain a stable average mis-*linking* probability when choosing K. Figure (B) shows the estimates of the drift velocity and diffusion constant including 3σ error bars as a result of the nested sampling algorithm and our Bayesian framework.

'easier' to recover compared to higher particle densities, where we can quickly mix up particles. In figure 3.3 (B) we can see that the parameter estimation works in general.

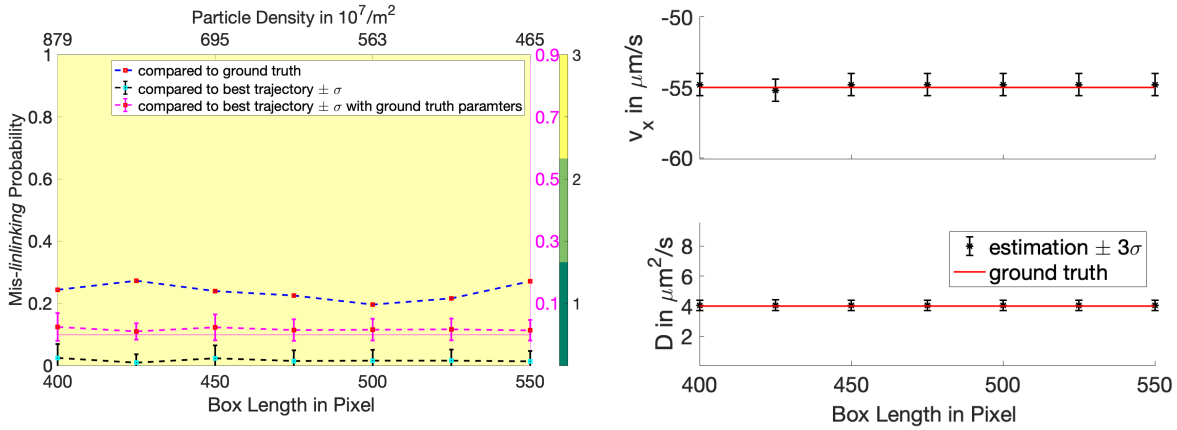
The estimated parameters including the error are mostly similar to the simulation input. We can estimate the diffusion constant especially well but obtain some outliers for very high particle densities. In figure 3.3 (A) we calculate the average mis-*linking* probability of each simulation using the best-fitting parameters (black graph with green squares). Comparing these results with the average mis-*linking* probability calculated with the simulation input parameters (magenta graph) and the actual mis-*linking* (blue graph with red squares) we see trends matching. We can obtain the actual mis-*linking* probability by comparing the ground truth trajectories to the proposed best set of trajectories. The average mis-*linking* probability using the simulation input (magenta graph) is very similar to the average mis-*linking* probability calculated with $(D^*, \vec{v}^*)$ and has therefore an offset of 0.1. We can see that we need more solutions of the Murty algorithm for higher particle densities to calculate a stable average mis-*linking* probability. If the particle density is increasing all three mis-*linking* probabilities are increasing as well.

3.3 Constrained Diffusive Motion in Nano-channels

In this section, we present results from one-dimensional Brownian motion, where the particles' motion is constrained by nano-channels. We simulate data close to the experimental data we have at hand to optimize the settings of our algorithms according to section 2.8, aiming to apply our methodology to the experimental data. A fairly 'easy' case of diffusive motion is when particles are trapped in nano-channels. Here we can lower the risk of particles interacting with each other and have the advantage of estimating the diffusion constant and the drift velocity in the x-direction only, as this diffusive motion is constrained in the y-direction. We simulate 240 particles doing five steps in different box sizes, for further simulation details we refer to section 3.1.

In figure 3.4 (B) we can see that the estimated diffusion constants D^* and drift velocities in the x-direction v_x^* are similar to the simulation input. That shows that the parameter estimation method works. Given those best estimates, we calculate the average mis-*linking* probabilities for each data set (black graph with green squares), as seen in figure 3.4 (A). The average mis-*linking* probabilities calculated with the ground truth parameters (magenta graph) are very similar to the mis-*linking* probabilities calculated with the best parameter estimate (D^*, v_x^*) and therefore has an offset of 0.1. We use the three best sets of trajectories to estimate those mis-*linking* probabilities. The rate of recovered trajectories (blue graph with red squares) does not match our mis-*linking* calculation and is higher than we estimate. Given these results, we move on to real experimental data.

We apply our methodology to a real experimental video with fluorescent molecules visible as white dots diffusing in nano-channels, as seen in figure 2.1. After detecting the dots in each image of this video with the ADCG algorithm we evaluate seven data sets of six consecutive images each. Once the dots in each nano-channel are determined we link those into trajectories and adjust the cost of non-assignment, which is set to



(A) (B)

FIGURE 3.4: **Mis-*linking* estimates compared to ground truth values & estimated model parameters for simulations close to experimental data.** The results shown here were obtained by simulating data which is as close as possible to the experimental data, which we like to analyze. We simulate 240 particles with a radius of 50nm taking five steps in 300nm wide nano-channels, which are confining the particles' motion in the y-direction. We consider different box sizes in our simulations and use periodic boundary conditions. We summarize all simulation parameters in table 3.1. Figure (A) shows that the average mis-*linking* probability in a data set calculated with the simulation input parameters $(D_{\text{sim}}, \vec{v}_{\text{sim}})$ (magenta with an offset of 0.1) and the estimated parameters (D^*, v_x^*) (black and green squares markers) is stable at around 0.03 for the tested particle densities. However, we underestimate the actual mis-*linking* probability (blue graph with red squares). The background color refers to the number of calculated solutions K, which is indicated by the color bar. We can see that we need to calculate three solutions to obtain stable average mis-*linking* probabilities. Figure (B) shows the estimation of the drift velocity in the x-direction and the diffusion constant including 3σ error bars as a result of the nested sampling algorithm and our Bayesian framework. The parameter estimation compared to the ground truth is promising and shows that our dimensional Brownian motion model with drift works. We note that the error of the diffusion constant is very low compared to earlier simulations in figure 3.3.

approximately a fifth of the 512 pixel long quadratic box. That means that we do not allow the Munkres algorithm to generate steps larger than that. In the data set we evaluate we have around 242 particles diffusing, which gives a particle density of $\approx 527 \cdot 10^7 \text{particles}/\text{m}^2$.

After generating trajectories of the experimental data sets we evaluate the best set of recovered trajectories with the mean displacement (MD) and the mean squared displacement (MSD), as seen in figure 3.5 (B) and (C) and in Appendix A. Here we present

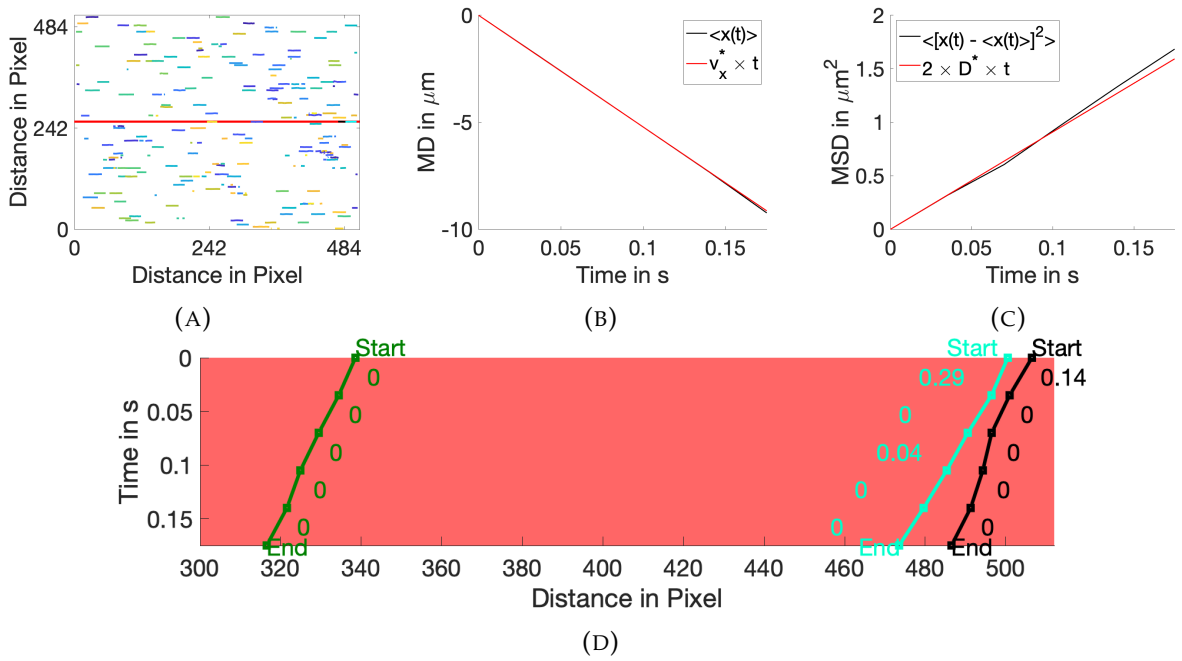


FIGURE 3.5: **Recovered trajectories of the best-fitting experimental data set, its mean displacement and mean squared displacement; including mis-*linking* probabilities for proposed steps in a specific nano-channel.** In figure (A) we display the recovered trajectories of experimental data as color traces, where fluorescent particles move constrained by nano-channels. This data set consists of six time-frames with a time of 0.035s in-between and approximately 242 particle. Based on the recovered steps in figure (A) we calculate the mean displacement (MD) $< x(t) >= v_{x,\text{fit}}t$ (black) and plot the estimated MD $v_x^* t$ (red) in figure (B). Figure (C) shows the mean squared displacement (MSD) $< [x(t) - < x(t) >]^2 >= 2D_{\text{fit}}t$ (black) and the expected values $2D^*t$ (red). Through the gradient of a linear fit in time we obtain a diffusion constant $D_{\text{fit}} \approx 4.5\mu\text{m}^2/\text{s}$ and drift velocity $v_{x,\text{fit}} \approx -53.2\mu\text{m/s}$. The red graphs are based on the best estimates $D^* = 4.6 \pm 0.8\mu\text{m}^2/\text{s}$ and $v_x^* = -52.2 \pm 0.8\mu\text{m/s}$ from the nested sampling approach. By comparing our estimates and the fitted parameter values we 'see' that the recovered trajectories behave as expected. Next, we calculate the four best solutions of Murty's algorithm and the mis-*linking* probabilities of the particles' most likely paths. In doing so we project the trajectories in the red nano-channel in figure (A) into a kymograph, as seen in figure (D). For illustration purposes, we display only trajectories, where we recover all five steps. We display the movement of the chosen particles in the x-direction and attach the mis-*linking* probabilities onto the arrows. The particles in the right part of figure (D) are physically close especially in the first step, here we obtain a high chance that the particles do not take the proposed step in reality.

the results of the data set, which has a MD and MSD most similar compared to our estimation, as seen in figure 3.5 (B) and (C). Based on the recovered steps in figure 3.5 (A) we calculate the MD = $< x(t) >= v_{x,\text{fit}}t$ and the MSD $< [x(t) - < x(t) >]^2 >= 2D_{\text{fit}}t$ and plot it in figure 3.5 (B) and (C) in black [14]. The MSD refers to the squared variance $\sigma^2 = 2D^*\Delta t$ and the MD to the mean $\mu = v_x^*\Delta t$ of our most likely one-dimensional Brownian motion model with drift. Both are linear in time and we can extract a drift velocity $v_{x,\text{fit}} \approx -53.2\mu\text{m/s}$ and diffusion constant $D_{\text{fit}} \approx 4.5\mu\text{m}^2/\text{s}$ through fitting a linear graph to the MD and MSD, which is very close to our best estimates $D^* = 4.6 \pm 0.8\mu\text{m}^2/\text{s}$ and $v_x^* = -52.2 \pm 0.8\mu\text{m/s}$. Then we can calculate the mis-*linking* probabilities using the four best sets of trajectories of Murty's approach of the Munkres' algorithm. The average mis-*linking* probability in the discussed data set is approximately 0.04 and matches quite well with the calculated average mis-*linking* probabilities for the simulation in figure 3.4 (A).

In figure 3.5 (D) we project the most likely trajectories of the particles in the red nano-channel seen in figure 3.5 (A) into a kymograph. For illustration purposes, we consider only trajectories, where we recover all five steps and display the motion of each particle in time versus movement in the x-direction. The particles follow the arrows on which we attach the mis-*linking* probabilities of each step. The first step of the particles on the right in cyan and black has a higher mis-*linking* probability compared to the other steps. Here the mis-*linking* probabilities express that the particles are very close and therefore have a high uncertainty of taking the best-proposed step.

Chapter 4

Discussion

In this chapter, we discuss the expectations and results of our proposed methodology in the case of constrained and unconstrained motion. We go through the linking and parameter estimation process, which works simultaneously. Giving the mis-*linking* probabilities is a separated process after our Brownian motion model with drift has been optimized and trajectories are generated. In general, we note that our methodology works but estimates higher mis-*linking* probabilities for boxes more densely packed with particles in nano-channels.

4.1 Recovering of Trajectories and Parameter Estimation

The estimation of parameters works very well for almost all tested particle densities, see figure 3.3 (B) and 3.5 (B). From this, we can conclude that our model is accurate enough to estimate the designated parameters of our physical model. The stopping condition of the nested sampling algorithm and also the sampling size is sufficient and provides good estimates of the designated parameters.

In contrast, the recovery rate of ground truth trajectories of particles in nano-channels in figure 3.4 (A) is not as good as we would expect from results of unconstrained diffusive motion in figure 3.3 (A).

This leads to the conclusion that our model is accurate enough to estimate the parameters but lacks in accuracy especially for higher particle density and constrained motion.

Another important aspect to mention is that depending on the choice of the diffusion constant D and the drift velocity $\vec{v}$ it is easier or harder to estimate the parameters correctly. We find that the absolute value of the mean in a specific direction defined by the drift velocity $|\mu| = |v * \Delta t|$ has to be bigger than the variance $\sigma = \sqrt{2D\Delta t}$. Otherwise the drift velocity does not affect the diffusive motion and we are very unlikely to get a good estimate of the drift velocity.

4.2 Calculating Mis-*linking* Probabilities

Here we compare the calculated average mis-*linking* probabilities of the data sets to the ground truth recovery rate of the trajectories. In theory, these should be very similar but show large discrepancies for the diffusive motion constrained with nano-channels. However, the average mis-*linking* probabilities for the unconstrained diffusive motion in figure 3.3 (A) have recovery rates and calculated mis-*linking* probabilities, which are following the same trends and matching most of the time.

In figure 3.4 (A) we calculate an average mis-*linking* probability which is too low compared to the actual recovery rate. One crucial step in this process is the setting of the number of calculated sets of trajectories K. We increase K and calculate another solution as long as the mis-*linking* probability is rising. However, it might be the case that we target a local maximum of the mis-*linking* probability and do not find the global maximum in dependency of K. Another fact is that if the particles are trapped in nano-channels they have a limited amount of alternative steps available as they cannot link to particles outside their nano-channel. This means if we only detect two particles in one nano-channel we will have no more than two possible solutions available, which can also decrease the calculated mis-*linking* probability in some steps.

In addition we calculated the mis-*linking* probability using the best estimate (black graph with green squares) and the ground truth parameters (magenta graph with an offset of 0.1) as seen in figure 3.3 (A) and 3.4 (A). These mis-*linking* probabilities are very similar and confirm the good estimate of the designated parameters.

Chapter 5

Summary

In this thesis, we presented a way to not only track and simultaneously estimate the diffusion constant and drift velocity of the investigated particles but to estimate the mis-*linking* probabilities of the best-proposed sets of trajectories.

We simulated the unconstrained and constrained diffusive motion of particles in a box with periodic boundaries and analyzed the generated data within our Bayesian Framework. Utilizing the Munkres' algorithm with Murty's approach we linked observed dots by maximizing our Brownian motion model with drift. With the help of the nested sampling algorithm we found the best sets of trajectory and the best-fitting set of parameters. The best-fitting parameters mostly agreed with the simulation input parameters in the case of the unconstrained and constrained diffusive motion. By comparing all of the obtained steps to the best-proposed solution and estimating the average mis-*linking* probability in each data set, we could assess our linking depending on the particle density. Knowing the ground truth trajectory recovery rate we slightly underestimated the mis-*linking* probability for constrained diffusive motion and high particle densities, but estimated drift velocities and diffusion constants successfully within the ground truth values. The estimation of mis-*linking* probabilities was more robust for motion which is not constrained.

Detecting fluorescent dots in experimental videos we used the ADCG algorithm and adapted our proposed methodology to apply it successfully on experimental data sets. Generating realistic trajectories we adjusted the length of allowed steps within one data set to compensate for missing or "extra" dots. In addition, we localized all nano-channels in the data sets and only linked the particles present in each nano-channel. In doing so, we do not allow a particle to move outside a nano-channel and increase computational speed as we deal with considerably smaller cost matrices. To verify our results we calculated the mean square displacement (MSD) and mean displacement (MD) of the recovered trajectories, which are proportional to the diffusion constant and drift velocity. Through a linear fit to the MSD and MD we could obtain another set of parameters and compared those with the estimated parameters. We calculated the mis-*linking* probabilities in the data set with the least discrepancy between the fitted and estimated parameters. Here we could get similar results compared to the simulation.

Chapter 6

Outlook

Here we discuss future challenges, applications, and potential improvements of our methods.

We achieve a great rate of trajectory recovery for unconstrained diffusive motion but lack that when working with constrained diffusive motion, in which particles are trapped in nano-channels. Therefore, we would like to recover trajectories more accurately or increase the estimation of the mis-*linking* probability, which in the unconstrained case is too low compared to the actual trajectory recovery rate. An exact idea on how to solve this issue is still missing.

The accuracy of the parameter estimation for unconstrained diffusive motion is another part we like to improve. Within our Bayesian framework, we use the nested sampling approach to estimate parameters. We could exchange this part of our methodology with other similar Monte Carlo approaches and compare the estimated parameters to the ground truth. Here we suggest a Markov chain Monte Carlo (MCMC) method such as the Metropolis-Hastings algorithm [15]. The Metropolis-Hastings algorithm samples posterior values of the parameters space and finds the best-fitting parameters by comparing them to the latest accepted posterior sample of a parameter set. Another approach, which is probably most interesting in complex multidimensional problems, is the tensor train (TT) representation of the parameter space [16, 17]. This approach decomposes and approximates the parameters space and saves computational time as the number of possible parameter combinations is significantly reduced.

Since we can work well with simulations of unconstrained diffusive motion, we now like to apply our methodology to unconstrained diffusive experiments. Here, we aim to extend the linking algorithm to deal with missing or extra dots similar to Jaqaman et al. [18]. In doing so, we could introduce another physical model. One appropriate physical model could describe the particle appearance or disappearance due to on and off switching of the fluorescent part of the molecule. Once this extra layer is implemented we could apply our methodology to various novel systems.

A suitable and interesting experiment to apply our methodology on is e.g. a T-cell experiment, where viruses are killed by immune cells (T-cells) after being detected by the cells' receptors. By labeling the viruses with a fluorescent dye and tracking them with our methodology we could gain knowledge on the binding rates of the receptors and viruses, which help us to understand the immune system's work. Here we most

likely have to work with more physical models, which leads to the estimation of multiple parameters. This includes e.g. two diffusion constants, one for the unbound and one for the bound state of the virus, and drift velocities in not only two but three dimensions, depending on the available experimental data.

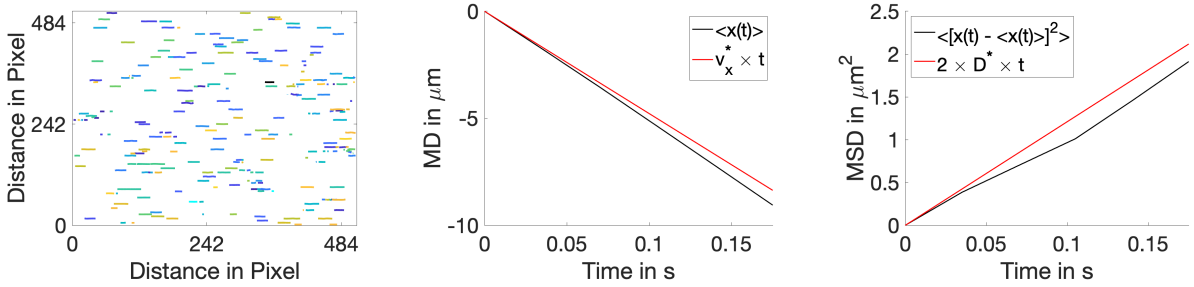
Another possible application is the optical reconstruction of barcodes of deoxyribonucleic acids (DNA). When stretched in silicon nano-channels, we can optically map fluorescent labeled DNA. DNA is a chain of molecules, which we encrypt with the four letters A C T G, and holds the blueprints of our life. Usually analyzed chemically, we could use the ADCG algorithm to first detect and then localize specific fluorescent markers. Here we would follow the dots and generate trajectories with our Bayesian framework. By averaging out any movements of the molecules in the DNA chain we obtain precise localization of each 'letter'. The main challenge in this process is to distinguish the different types of molecules, as they are physically close and blurry due to background noise in the data [4, 19]. When successfully applied we can construct DNA barcodes faster than ever and outperform existing methods.

The last utilization of our Bayesian framework which we like to mention is the characterization of nano- and micro-fluids. By determining the motion describing parameters of tracked particles in the fluid we can extract, for instance, the particle's size through the estimated drift velocity and diffusion constant in combination with Stokes law. On the other hand, if our Bayesian framework is implemented using population-based models instead of focusing on single particles, we can quantify various contents in the fluid. Here, one could analyze the components of blood samples or other samples of patients to gain important information about the patients' health status within minutes. This method is not so time-intensive and does not require as much extensive effort in the analysis compared to existing methods.

Appendix A

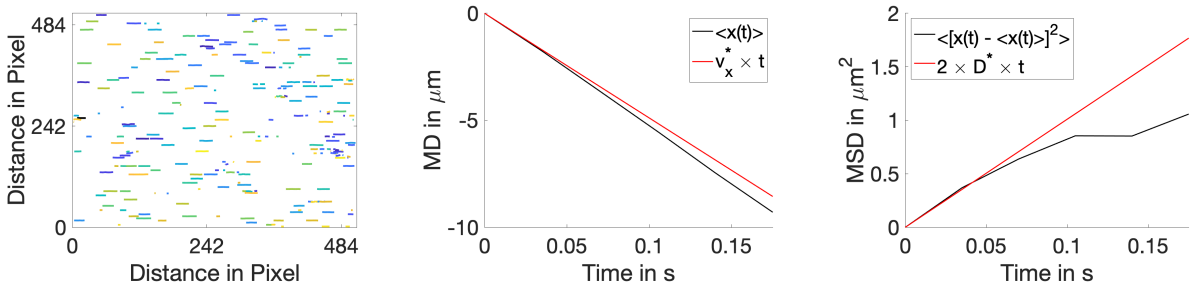
Supplementary Figures

Here we present the results of the experimental data sets we did not present in section 3.3. For each of the data sets we calculate the mean squared displacement (MSD) $< [x(t) - < x(t) >]^2 >= 2D_{\text{fit}}t$ (C) and mean displacement (MD) $< x(t) >= v_{x,\text{fit}}t$ (B) based on the recovered steps (A). Through fitting we get the parameter values for the drift velocity in the x-direction $v_{x,\text{fit}}$ and the diffusion constant D_{fit}. We present the data set with the smallest difference of the estimated parameters (D^*, v_x^*), using our Bayesian framework, and the fitted parameters $(D_{\text{fit}}, v_{x,\text{fit}})$, in figure 3.4. The here shown data sets have larger differences and do not behave linearly as we expect and therefore do not represent a suitable result.



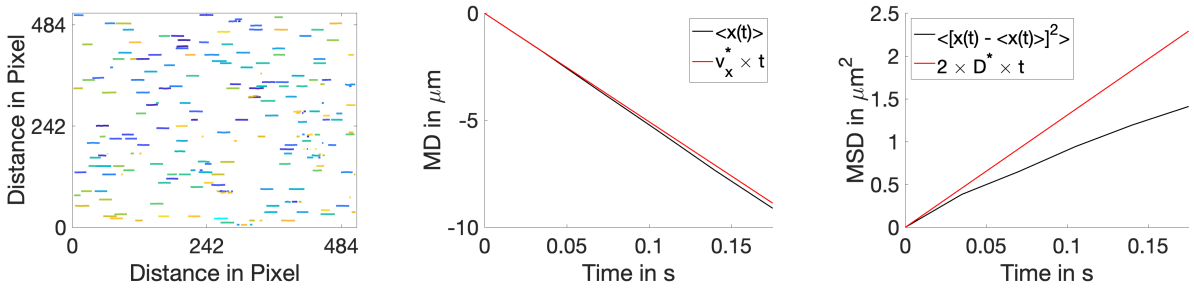
(A) (B) (C)

FIGURE A.1: **Quick assessment of recovered steps in experimental data set 1 with mean distance and mean squared distance.** Based on the recovered steps of ≈ 241 particles taking five steps in one experimental data set as seen in figure (A) we calculate the mean squared displacement (MSD) $< [x(t) - < x(t) >]^2 >= 2D_{\text{fit}}t$ in figure (C), after calculating the mean displacement (MD) $< x(t) >= v_{x,\text{fit}}t$. The calculated graphs are black and through fitting to the calculation we can obtain the diffusion constant $D_{\text{fit}} \approx 5.7\mu\text{m}^2/\text{s}$ and drift velocity $v_{x,\text{fit}} \approx -50.3\mu\text{m}/\text{s}$ when comparing with the gradient. The red graphs are based on the best estimates $D^* = 6.1 \pm 0.9\mu\text{m}^2/\text{s}$ and $v_x^* = -48 \pm 1\mu\text{m}/\text{s}$ from the nested sampling approach. By comparing the red and the black graph we can 'see' how well the real data fits the expectations of the physical model. If the MD and MSD behave linearly we can use this data set for further calculations. Here we decide not to do so as experimental data set 3 in figure **??** fits better to our expectations.



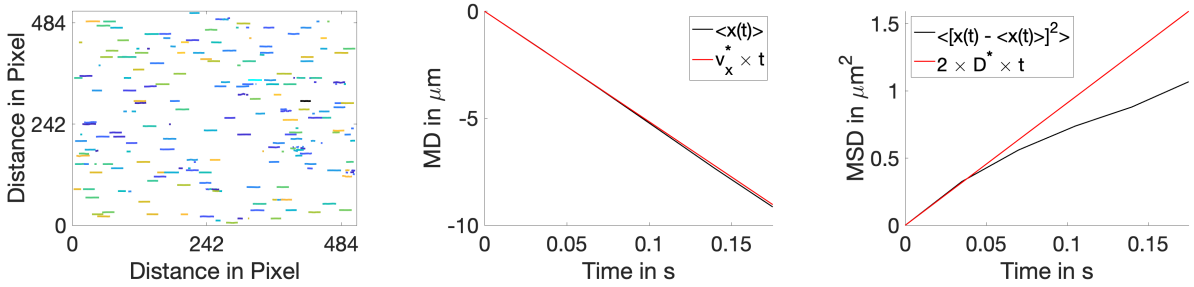
(A) (B) (C)

FIGURE A.2: **Quick assessment of recovered steps in experimental data set 2 with mean distance and mean squared distance.** Based on the recovered steps of ≈ 236 particles taking five steps in one experimental data set as seen in figure (A) we calculate the mean squared displacement (MSD) $< [x(t) - < x(t) >]^2 >= 2D_{\text{fit}}t$ in figure (C), after calculating the mean displacement (MD) $< x(t) >= v_{x,\text{fit}}t$. The calculated graphs are black and through fitting to the calculations we can obtain the diffusion constant $D_{\text{fit}} \approx 2.9\mu\text{m}^2/\text{s}$ and drift velocity $v_{x,\text{fit}} \approx -53.5\mu\text{m}/\text{s}$ when comparing to the gradient. The red graphs are based on the best estimates $D^* = 5.1 \pm 0.8\mu\text{m}^2/\text{s}$ and $v_x^* = -48.9 \pm 0.9\mu\text{m}/\text{s}$ from the nested sampling approach. By comparing the red and the black graph we can 'see' how well the real data fits the expectations of the physical model. If the MD and MSD behave linearly we can use this data set for further calculations. Here we decide not to do so as the MSD does not graduate linear in time.



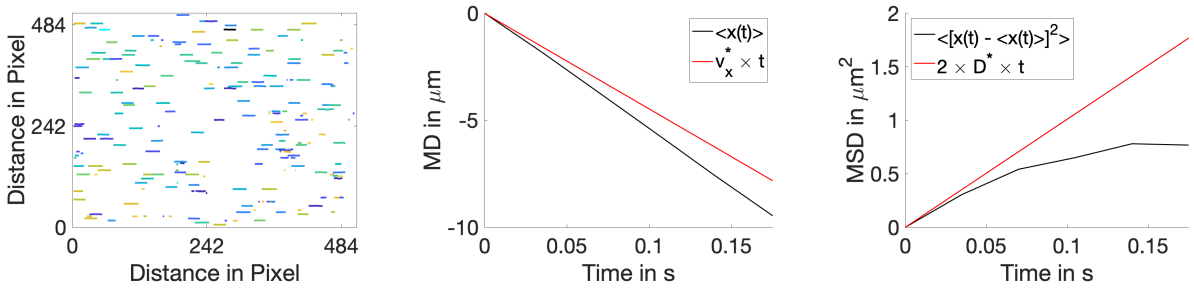
(A) (B) (C)

FIGURE A.3: **Quick assessment of recovered steps in experimental data set 4 with mean distance and mean squared distance.** Based on the recovered steps of ≈ 234 particles taking five steps in one experimental data set as seen in figure (A) we calculate the mean squared displacement (MSD) $< [x(t) - < x(t) >]^2 >= 2D_{\text{fit}}t$ in figure (C), after calculating the mean displacement (MD) $< x(t) >= v_{x,\text{fit}}t$. The calculated graphs are black and through fitting to the calculations we can obtain the diffusion constant $D_{\text{fit}} \approx 2.4\mu\text{m}^2/\text{s}$ and drift velocity $v_{x,\text{fit}} \approx -52.6\mu\text{m}/\text{s}$ when comparing to the gradient. The red graphs are based on the best estimates $D^* = 6.6 \pm 1.2\mu\text{m}^2/\text{s}$ and $v_x^* - 50.7 \pm 0.1\mu\text{m}/\text{s}$ from the nested sampling approach. By comparing the red and the black graph we can 'see' how well the real data fits the expectations of the physical model. If the MD and MSD behave linearly we can use this data set for further calculations. Here we decide not to do so as the MSD does not graduate linear in time.



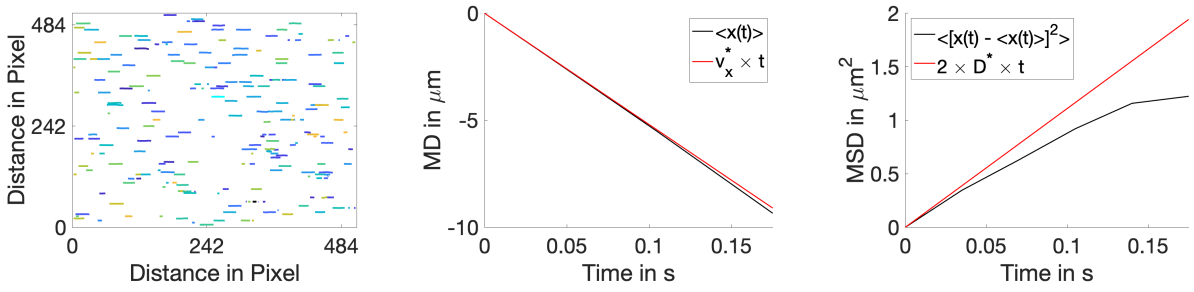
(A) (B) (C)

FIGURE A.4: **Quick assessment of recovered steps in experimental data set 5 with mean distance and mean squared distance.** Based on the recovered steps of ≈ 236 particles taking five steps in one experimental data set as seen in figure (A) we calculate the mean squared displacement (MSD) $< [x(t) - < x(t) >]^2 >= 2D_{\text{fit}}t$ in figure (C), after calculating the mean displacement (MD) $< x(t) >= v_{x,\text{fit}}t$. The calculated graphs are black and through fitting to the calculations we can obtain the diffusion constant $D_{\text{fit}} \approx 2.8\mu\text{m}^2/\text{s}$ and drift velocity $v_{x,\text{fit}} \approx -53\mu\text{m}/\text{s}$ when comparing to the gradient. The red graphs are based on the best estimates $D^* = 4.6 \pm 0.8mu\text{m}^2/\text{s}$ and $v_x^* = -51.6 \pm 0.9\mu\text{m}/\text{s}$ from the nested sampling approach. By comparing the red and the black graph we can 'see' how well the real data fits the expectations of the physical model. If the MD and MSD behave linearly we can use this data set for further calculations. Here we decide not to do so as the MSD does not graduate linear in time.



(A) (B) (C)

FIGURE A.5: **Quick assessment of recovered steps in experimental data set 6 with mean distance and mean squared distance.** Based on the recovered steps of ≈ 228 particles taking five steps in one experimental data set as seen in figure (A) we calculate the mean squared displacement (MSD) $< [x(t) - < x(t) >]^2 >= 2D_{\text{fit}}t$ in figure (C), after calculating the mean displacement (MD) $< x(t) >= v_{x,\text{fit}}t$. The calculated graphs are black and through fitting to the calculations we can obtain the diffusion constant $D_{\text{fit}} \approx 1.7\mu\text{m}^2/\text{s}$ and drift velocity $v_{x,\text{fit}} \approx -53.2\mu\text{m}/\text{s}$ when comparing to the gradient. The red graphs are based on the best estimates $D^* = 5.1 \pm 0.9\mu\text{m}^2/\text{s}$ and $v_x^* = -44.7 \pm 0.9\mu\text{m}/\text{s}$ from the nested sampling approach. By comparing the red and the black graph we can 'see' how well the real data fits the expectations of the physical model. If the MD and MSD behave linearly we can use this data set for further calculations. Here we decide not to do so as the MSD does not graduate linear in time.



(A) (B) (C)

FIGURE A.6: **Quick assessment of recovered steps in experimental data set 7 with mean distance and mean squared distance.** Based on the recovered steps of ≈ 246 particles taking five steps in one experimental data set as seen in figure (A) we calculate the mean squared displacement (MSD) $< [x(t) - < x(t) >]^2 >= 2D_{\text{fit}}t$ in figure (C), after calculating the mean displacement (MD) $< x(t) >= v_{x,\text{fit}}t$. The calculated graphs are black and through fitting to the calculations we can obtain the diffusion constant $D_{\text{fit}} \approx 2.7\mu\text{m}^2/\text{s}$ and drift velocity $v_{x,\text{fit}} \approx -53.5\mu\text{m}/\text{s}$ when comparing to the gradient. The red graphs are based on the best estimates $D^* = 6 \pm 1\mu\text{m}^2/\text{s}$ and $v_x^* = -52.0 \pm 0.9\mu\text{m}/\text{s}$ from the nested sampling approach. By comparing the red and the black graph we can 'see' how well the real data fits the expectations of the physical model. If the MD and MSD behave linearly we can use this data set for further calculations. Here we decide not to do so as the MSD does not graduate linear in time.

Bibliography

[1] Antony Lee, Konstantinos Tsekouras, Christopher Calderon, et al. "Unraveling the thousand word picture: an introduction to super-resolution data analysis". In: *Chemical reviews* 117.11 (2017), pp. 7276–7330.

[2] Daniel Sage, Thanh-An Pham, Hazen Babcock, et al. "Super-resolution fight club: assessment of 2D and 3D single-molecule localization microscopy software". In: *Nature methods* 16.5 (2019), pp. 387–395.

[3] Nicholas Boyd, Geoffrey Schiebinger, and Benjamin Recht. "The alternating descent conditional gradient method for sparse inverse problems". In: *SIAM Journal on Optimization* 27.2 (2017), pp. 616–639.

[4] Dolev Illouz. "Multiple-Emitter Super-resolution Imaging using the Alternating Descent Conditional Gradient Method". Bachelor's Thesis. University of Lund, Sweden: Computational Biology, Biological Physics Department of Astronomy, and Theoretical Physics, 2020.

[5] Nicolas Chenouard, Ihor Smal, Fabrice De Chaumont, et al. "Objective comparison of particle tracking methods". In: *Nature methods* 11.3 (2014), pp. 281–289.

[6] Devinderjit Sivia and John Skilling. *Data analysis: a Bayesian tutorial*. OUP Oxford, 2006.

[7] James Munkres. "Algorithms for the assignment and transportation problems". In: *Journal of the society for industrial and applied mathematics* 5.1 (1957), pp. 32–38.

[8] Harold W Kuhn. "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[9] Katta G Murty. "Letter to the editor—An algorithm for ranking all the assignments in order of increasing cost". In: *Operations research* 16.3 (1968), pp. 682–687.

[10] Rob Phillips, Jane Kondev, Julie Theriot, et al. *Physical biology of the cell*. Garland Science, 2012.

[11] Radek Erban, Jonathan Chapman, and Philip Maini. "A practical guide to stochastic simulations of reaction-diffusion processes". In: *arXiv preprint arXiv:0704.1908* (2007).

[12] Jens Krog, Lars H Jacobsen, Frederik W Lund, et al. "Bayesian model selection with fractional Brownian motion". In: *Journal of Statistical Mechanics: Theory and Experiment* 2018.9 (2018), p. 093501.

[13] John Skilling. “Nested sampling for general Bayesian computation”. In: *Bayesian analysis* 1.4 (2006), pp. 833–859.

[14] Xin Bian, Changho Kim, and George Em Karniadakis. “111 years of Brownian motion”. In: *Soft Matter* 12.30 (2016), pp. 6331–6346.

[15] W Keith Hastings. “Monte Carlo sampling methods using Markov chains and their applications”. In: (1970).

[16] Ivan V Oseledets. “Tensor-train decomposition”. In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2295–2317.

[17] Colin Fox, Sergey Dolgov, Malcolm EK Morrison, et al. “Grid methods for Bayes-optimal continuous-discrete filtering and utilizing a functional tensor train representation”. In: *Inverse Problems in Science and Engineering* 29.8 (2021), pp. 1199–1217.

[18] Khuloud Jaqaman, Dinah Loerke, Marcel Mettlen, et al. “Robust single-particle tracking in live-cell time-lapse sequences”. In: *Nature methods* 5.8 (2008), pp. 695–702.

[19] Jonathan Jeffet, Asaf Kobo, Tianxiang Su, et al. “Super-resolution genome mapping in silicon nanochannels”. In: *ACS nano* 10.11 (2016), pp. 9823–9830.