Title

Extracting fast subpopulations from fragmentary live cell single-particle trajectories

Author

Heckert, Alec Basil

Publication Date

2021

Peer reviewed|Thesis/dissertation

Extracting fast subpopulations from fragmentary live cell single-particle trajectories

By

Alec B Heckert

Dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy
in
Molecular & Cell Biology
in the
Graduate Division
of the
University of California, Berkeley

Thesis Committee:
Professor Xavier Darzacq, co-chair
Professor Robert Tjian, co-chair
Professor Hernan Garcia
Professor David Schaffer

Spring 2021

Abstract

Extracting fast subpopulations from fragmentary live cell single-particle trajectories

by

Alec B Heckert

Doctor of Philosophy in Molecular & Cell Biology

University of California, Berkeley

Professors Xavier Darzacq & Robert Tjian, Co-Chairs

Stroboscopic photoactivated single particle tracking (spaSPT) relies on stochastic labeling to isolate the paths of individual fluorophores and can provide information about the behavior of biological macromolecules in their native cellular environment. Existing spaSPT modalities generate large numbers of short trajectories, each representing a fragment of an individual emitter's path. When interpreting this data through the lens of diffusion models, it is essential to account for the fragmentary nature of trajectories, experimental biases arising from the imaging geometry, and our ignorance about the correct underlying diffusion model. In this thesis, we describe several methods for interpretation of spaSPT data that estimate the number and characteristics of mobility states in spaSPT data while accounting for known experimental artifacts. We explore the uses and limitations of these models on simulated and experimental datasets.

In the final chapter, we apply these methods to study the competitive chromatin binding in the type II nuclear receptors (T2NRs). T2NRs are a class of ligand-activated transcription factors that require heterodimerization with a common factor, the retinoid X receptor (RXR), to bind chromatin and regulate target genes. Because all T2NRs must dimerize with a common pool of RXR, competition between individual T2NRs may limit access to the bound state, a mechanism has been proposed to underlie the inactivation of the wildtype retinoic acid receptor alpha (RARA) in the presence of RARA fusion proteins that occur in acute promyelocytic leukemia (APL). We apply spaSPT to measure the effects of RARA fusion proteins on the chromatin binding of endogenously tagged RARA and RXR. Using tools developed in the previous chapters, we find that RARA fusion proteins act as stronger competitors for dimerization with RXR than wildtype T2NRs, and are also apparently exempt from autoregulation of RARA concentration. Together, these results provide new insights into the interdependence of T2NR gene regulation.

Contents

List of figures

List of algorithms

List of tables

Conventions

This is a list of symbols and conventions used throughout this thesis.

symbol(s)	definition
$A \circ B$	Hadamard product of matrices A and B
$A * B$	Convolution of A with B
$\mathbb{X}$	a set of trajectories, not necessarily the same length
$\nabla\nabla^T f$	Hessian matrix of a multivariate function f
$\mathcal{F}[f]$	Fourier transform of a function f with respect to all of its variables
$\mathcal{F}_x[f]$	Fourier transform of a function f with respect to x
$\mathcal{L}[f]$	Laplace transform of a function f
$\mathcal{R}[f]$	Radon transform of a function f

Probability

When dealing with probability distributions, we use the following conventions:

- $f_X(x)$: probability density function for a random variable X
- $F_X(x)$: cumulative distribution function for a random variable X
- $\phi_X(x)$: characteristic function for a random variable X

We use $X \sim f$ to mean that the random variable X is distributed according to the density function f. For instance, $X \sim \mathcal{N}(0, \nu^2)$ means that X is distributed according to a normal distribution with zero mean and variance ν^2.

We use $\mathbb{E}[X]$ to mean the expected value of a random variable X, and $\mathbb{E}[X \mid Y]$ to mean the expected value of X given Y.

We use a subscript on the expectation to represent taking the expectation with respect to a subset of the parameters for a joint distribution. For example, if X and Y are continuous random variables,

$$\mathbb{E}_X[g(X, Y)] = \int_{\mathbb{R}} g(x, y)\, f_{X,Y}(x, y)\, \mathrm{d}x$$

For some common distributions, we use the following notation:

$\mathcal{N}(\mu, \nu^2)$ is the univariate normal distribution with mean μ and variance ν^2, so that if $X \sim \mathcal{N}(\mu, \nu^2)$, then

$$f_X(x) = \frac{1}{\sqrt{2\pi\nu^2}} \exp\left(-\frac{(x-\mu)^2}{2\nu^2}\right)$$

$\mathcal{N}(\boldsymbol{\mu}, C)$ is the multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance C, so that if $\boldsymbol{\mu} \in \mathbb{R}^m$, $\mathbf{C} \in \mathbb{R}^{m \times m}$, and $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$, then

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T C^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{(2\pi)^{m/2}(\det(C))^{\frac{1}{2}}}$$

Gamma(α, β) is the gamma distribution, so that if $X \sim$ Gamma(α, β), then

$$f_X(x) = \begin{cases} \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

InvGamma(α, β) is the inverse gamma distribution, so that if $X \sim$ InvGamma(α, β), then

$$f_X(x) = \begin{cases} \frac{\beta^\alpha e^{-\beta/x}}{\Gamma(\alpha) x^{\alpha+1}} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Dirichlet$(\mathbf{n})$ is the Dirichlet distribution, so that if $\mathbf{n}, \boldsymbol{\tau} \in \mathbb{R}^K$ and $\boldsymbol{\tau} \sim$ Dirichlet$(\mathbf{n})$, then

$$f_{\boldsymbol{\tau}}(\boldsymbol{\tau}) = \frac{1}{\mathbf{B}(\boldsymbol{\tau})} \prod_{j=1}^{K} \tau_j^{n_j - 1}$$

where $\mathbf{B}(\mathbf{n})$ is the multivariate beta function, given by

$$\mathbf{B}(\mathbf{n}) = \frac{\Gamma(n_1) \cdots \Gamma(n_K)}{\Gamma(n_1 + \ldots + n_K)}$$

Spatial coordinates

We will generally take the following unless otherwise specified:

- $\mathbf{R} = (X_1, ..., X_m)^T$ is the vector jump of a particle in m dimensions. That is, it represents the spatial coordinates of a particle that starts out at the origin after time t.
- $\mathbf{R}^2 = (X_1^2, ..., X_m^2)^T$ is the corresponding squared vector jump.
- $S = R^2 = \sum_{i=1}^{m} \mathbf{R}_i^2$ is the scalar squared radial jump of the particle.
- $R = \sqrt{S}$ is the scalar radial jump.
- R_1 denotes the scalar radial jump in 1 dimension, R_2 denotes the scalar radial jump in 2 dimensions, and so on.

Fourier transforms

Several parts of the thesis use the characteristic function from probability theory. To keep the Fourier transform consistent with the definition of the characteristic function, we use the Fourier transform defined by

$$\tilde{f}(\mathbf{k}) = \mathcal{F}[f] = \int_{-\infty}^{+\infty} f(\mathbf{x}) e^{i\mathbf{k}^T\mathbf{x}} d\mathbf{x}$$

which corresponds to the inverse transform

$$f(\mathbf{x}) = \mathcal{F}^{-1}\left[\tilde{f}\right] = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \tilde{f}(\mathbf{k}) e^{-i\mathbf{x}^T\mathbf{k}} d\mathbf{k}$$

In this way, the characteristic function of a random variable X can be written simply as the Fourier transform of its PDF:

$$\phi_X(k) = \mathbb{E}\left[e^{ikX}\right] = \int_{-\infty}^{+\infty} f_X(x) e^{ikx} dx = \mathcal{F}[f_X](k)$$

As usual, for the Laplacian we have

$$\mathcal{F}\left[\nabla^2 f\right] = -|\mathbf{k}|^2 \tilde{f}(\mathbf{k})$$

despite the change in the sign of the exponent relative to the usual Fourier transform.

Chapter 1

Introduction

Biological processes are driven by the movements and interactions of discrete molecules. Because cells are mixtures of thousands of different molecular species, it has been historically challenging to isolate the role of any single component.

Fluorescence microscopy has proven to be a powerful paradigm to overcome this challenge. Genetically encoded fluorescent or photoconvertible proteins [1] [18] and biologically orthogonal dyes [12] [14] can be used to selectively visualize subpopulations of components inside cells. These labeling strategies provide the basis for a collection of techniques that measure the dynamics of single molecular species in live cells. Such techniques include fluorescence recovery after photobleaching (FRAP) [16], fluorescence correlation spectroscopy (FCS) [17], Förster resonance energy transfer (FRET), small molecule biosensors, and fluorescent single particle tracking [5] [6] [7].

One such approach is the use of photoconvertible, photoswitchable, or photoactivatable fluorophores, a strategy descended from fixed-cell stochastic labeling approaches including photoactivated localization microscopy (PALM) [3] and stochastic optical reconstruction microscopy (STORM) [4]. While chemically diverse, these techniques share the common strategy of limiting coincidentally fluorescent molecules per cell to a bare handful. At these low densities, fluorophores are sufficiently separated to identify individual molecules. Fast detectors are then used to track the motion of emitters between frames [8] with short pulses of excitation light to limit motion blur [21]. Individual fluorophores bleach quickly, so these methods cycle between imaging active fluorophores and renewing the population of active fluorophores. The result is thousands to tens of thousands of trajectories per cell, each generated by the motion of a single fluorophore. Together, these innovations have enabled the application of SPT to intracellular settings with fast-moving subpopulations [9] [10] [11] [23]. We refer to this class of techniques as *stroboscopic photoactivated single particle tracking* (spaSPT).

In this thesis, we examine methods to identify and extract subpopulations of trajectories with distinct mobility characteristics from spaSPT datasets. These methods are prerequisite or related to several of the more common types of information accessible via spaSPT, including:

1. *Viscosities*. By measuring the motion of a tracer fluorescent protein with known Stokes radius, spaSPT can be used to estimate spatially-resolved viscosities, providing information about subcellular environments [19].

2. *Active vs. passive transport*. The statistical characteristics of trajectories can be used to identify when a particle is moving in a directed manner [20]. *Fractional Brownian motion* (explored later in this thesis) provides a powerful statistical framework for testing these hypotheses.

3. *Existence and occupancies of distinct mobility states*. By resolving sets of trajectories into subpopulations with distinct mobility characteristics, spaSPT can provide information about the fraction of particles involved in different activities. These techniques become particularly useful when coupled with domain deletions and mutations that can help assign the observed subpopulations to specific molecular functions [23] [60].

4. *Kinetics of state transitions*. By identifying transitions between states with distinct mobility characteristics, spaSPT can be used to measure rates of conversion between these states [50] [21].

5. *Barriers to molecular motion*. Trajectories provide information about which parts of the cell are accessible from other parts. The set of all trajectories in a cell can be used to identify barriers to motion, or can be used to test hypotheses about whether a given subcellular feature presents a barrier of motion [24].

6. *Distances between components*. The ability of spaSPT to resolve absolute distances between molecular components at super-optical resolutions has proven particularly useful when studying the dynamics of chromatin looping [25].

7. *Confinement and molecular crowding*. There have been attempts to apply spaSPT to measure molecular crowding, although these applications are in their infancy [26].

These aspects of spaSPT have attracted decades of attention from the biophysics community [27], particularly in the classic problems of membrane receptor dynamics [22] transcription factor target search [28]. As a result, the spaSPT field

benefits from a wealth of theory relating underlying physical structure of biological systems to their dynamics. However, comparatively little attention has been focused on robust and scalable statistical procedures to extract this information from actual spaSPT datasets. In some cases, the techniques applied to spaSPT data today vary little from the experiments of Jean Perrin over a century ago [29]. Analysis is often performed on small, handpicked datasets with manual oversight over each step, posing challenges for scaling up the assay.

As a demonstration of the issues that confront the spaSPT practitioner, we briefly highlight five challenges that are discussed later in the thesis.

Challenge 1: Localization error

spaSPT relies on statistical procedures inherited from PALM and STORM to estimate the position of fluorescent emitters at subpixel resolution, given the distribution of light they present to a detector. In fixed-cell PALM/STORM, one can reasonably assume that this distribution is given by the microscope's point spread function (PSF). This is *not* the case for mobile emitters, which generate a distribution (colloquially, a "spot") produced by the convolution of the PSF with the path of the emitter.

This raises two issues. First, since the true path of an emitter is unknown, "localization error" - the deviation of the estimated position from the particle's true position - is undefined in this context, since there is no single true position. Whether a point estimate is supposed to report on the center of mass of an emitter's path or some other statistic is often left unsaid. As a result, reported estimates of "localization error" are entirely dependent on whatever method was used to measure it.

Second, because the path of each emitter is different and cannot be predicted, the error associated with estimating a mobile emitter's position is higher than for immobile emitters. Exactly how much higher is usually unknown. The typical way to estimate localization error in PALM/STORM - measuring the variance in the position of an immobile probe - does not accurately reflect the error associated with a mobile molecule. In other words, localization error is a function of the mobility characteristics of the emitter [30] [31] [32].

We are aware of neither a commonly accepted convention in the spaSPT field to measure the localization error of mobile emitters, nor a comprehensive comparison of the error with which different PALM/STORM localization methods identify the "position" (say, the integrated path center of mass) of mobile emitters. (In Chapter 2 of this thesis, we propose methods that could form the basis for such a comparison.)

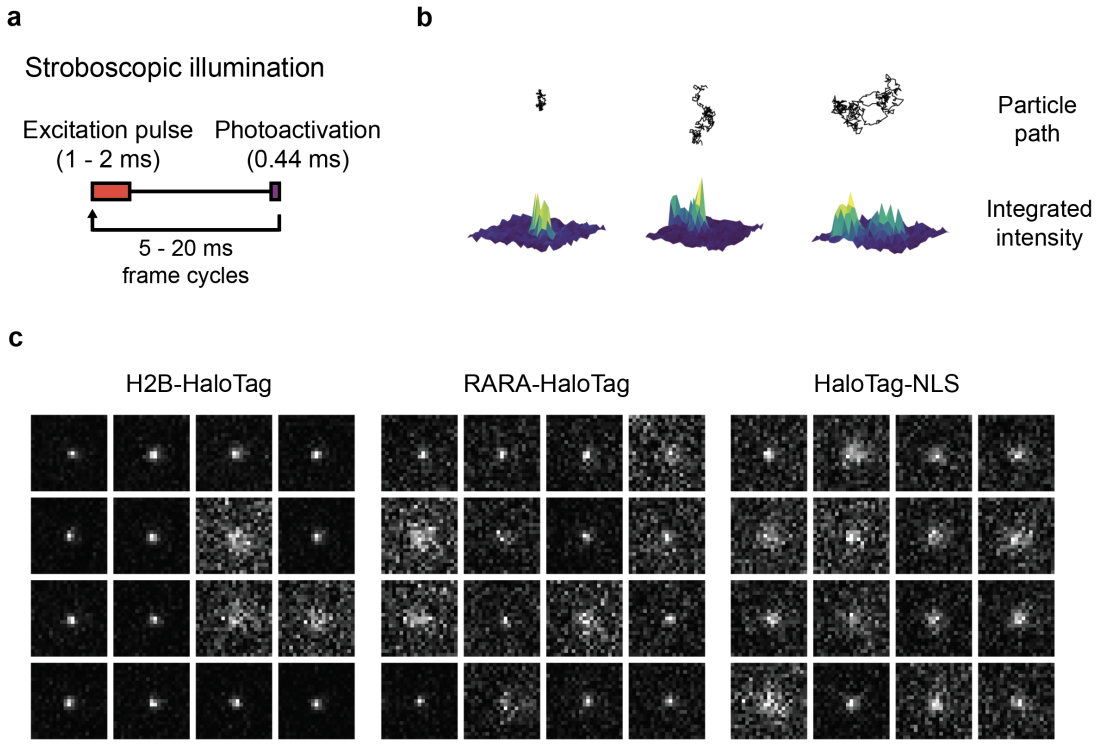

Figure 1.1: Illustration of the role of localization error/motion blur on the spaSPT measurement. (a) Temporal cycle for stroboscopic illumination, intended to limit the effects of motion blur. The excitation pulse is concentrated in a 1-2 ms period at the beginning of each integration period, while photoactivation (if relevant) is performed during the camera transition time. (b) Some simulated paths of particles and the integrated intensities on a hypothetical camera. The faster a particle moves, the broader the apparent spot it presents. (c) Observed spots for three different proteins. All proteins were conjugated to HaloTag, labeled with photoactivatable PA-JF549 dye [15], and tracked with 7.48 ms frame intervals with 1.5 ms pulse widths. Sixteen spots were randomly selected from each dataset.

At the same time, the issue of localization error assumes even greater importance for downstream analysis in spaSPT than in fixed cell PALM/STORM. Most of the time, spaSPT analysis involves making estimates of motion from trajectories. Due to localization error, even immobile particles appear to move. When analyzing spaSPT data in terms of single jumps between frames, this "apparent" motion due to error has characteristics indistinguishable from regular Brownian motion. Worse, when incorporating information across multiple frames, localization error presents signatures of subdiffusion - usually one of the primary characteristics of molecular motion on which spaSPT is supposed to report. Even regular Brownian motion, a quintessential Markov process, becomes non-Markovian in the presence

of localization error. These effects are particularly dangerous for methods such as angular distributions. (We investigate these effects in section 3.2.)

In this thesis, while we develop most of our estimators with explicit consideration of localization error, it remains for future work to determine robust ways to jointly estimate localization error and model parameters for multi-state diffusion models in spaSPT experiments.

Challenge 2: Observation geometry

To collect a sufficient number of photons from individual fluorescent emitters for detection, typical spaSPT setups rely on high numerical aperture (NA) objectives with short depth of field ("focal depth") - often as little as 500-1000 nm [23]. At the same time, the interval between frames must be constrained to a few milliseconds to observe the high speed of molecular motion. These combined requirements for high NA and fast acquisition mean that most spaSPT setups measure motion in a single, thin 2D plane, as there are currently no widely adopted methods to gather *z*-stacks at the speeds requisite for these experiments. Multi-focal plane microscopy presents a promising avenue to escape this limit [33], but remains a specialized technique maintained by a small handful of laboratories.

The diffusion coefficient for free HaloTag or GFP molecules inside the cell is often measured between 20 and 60 $\mu m^2\ s^{-1}$. If observed at 5 ms frame intervals, this range corresponds to mean 2D radial jumps of 560 nm to 970 nm. These jumps are comparable to the focal depth itself. Many or most of them are lost because both endpoints do not fall within focus. In contrast, a slow-moving molecule with a diffusion coefficient 1.0 $\mu m^2\ s^{-1}$ has a mean radial jump of 125 nm, meaning that most of its jumps will be observed. In a population comprised of half slow-moving, half fast-moving molecules, the majority of observed jumps will be collected from emitters the slow state, leading to an inherent bias in the estimation of different states [75] [59] [60]. In this thesis, we term this situation *defocalization bias*.

Accounting for defocalization bias is not trivial. Existing methods are based on Monte Carlo presimulations of the specific experimental conditions in question, and are limited to the special case of regular Brownian motion. In chapter 4, we present numerical procedures to account for defocalization bias to arbitrary precision without simulation and for much broader categories of motion. These methods are based on the Green's function for the motion (in the case of Markov processes) or on the model covariance matrix (for fractional Brownian motion).

State biases are not the only issue presented by short focal depth. Probably the most important issue is that, because trajectories only make short transits through

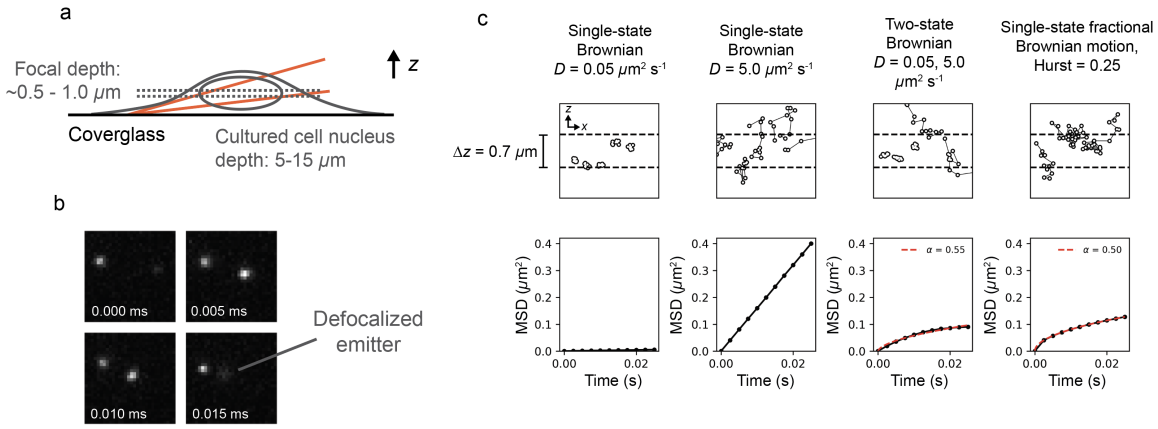

Figure 1.2: Demonstration of the effect of defocalization on the analysis of spaSPT data. (a) Schematic of the size of a typical spaSPT focal depth compared to a nucleus in mammalian cell culture. The red lines indicate the excitation profile under HiLo illumination [13]. The focal depth, rather than the width of the excitation sheet, is the limiting factor for observation. (b) Several frames from an spaSPT movie, showing the loss of an emitter due to defocalization. (c) Examination of the consequence of defocalization for MSD analysis of simulated data. In the left two subplots, a homogeneous set of trajectories were simulated, resulting in straight MSDs as expected for Fickian diffusion. However, when the states are mixed (in the middle-right subplot), defocalization of the faster state results in an apparent sublinear MSD, similar to true subdiffusive models (far right subplot). In order to distinguish true subdiffusion from the presence of multiple diffusing states, it is necessary to take into account the effect of defocalization.

the focal volume, trajectory lengths for moving particles are often limited to a few frames (as few as 3-4 on average in our setup). Defocalization is actually more consequential for trajectory length than photobleaching in a typical fast tracking experiment (Fig. 1.3). The extremely limited set of points for most trajectories poses serious problems for analysis methods based on analyzing temporal correlations in molecular behavior, such as hidden Markov models.

Finally, for some categories of motion, jumps in 3D space may not be separable into *x*, *y*, and *z* components. This is the case, for example, in every kind of Levy flight except for regular Brownian motion. For these types of motion, truncating the observed jumps in *z* by imposing a short focal depth actually affects the distributions of *xy* jumps presented by the particle. In chapter section 3.1 we will see that this kills the Markov property of Levy flights. The strongest effects impact the same category of long jumps that are usually considered to be characteristic of the Levy flight stability parameter.

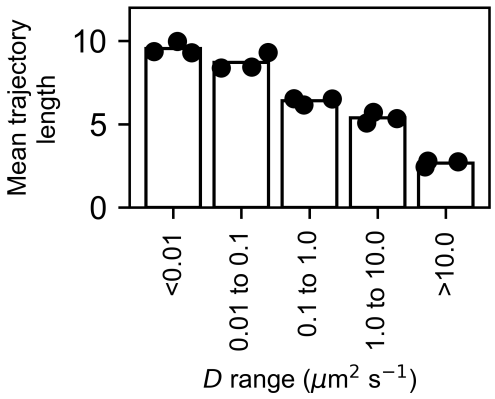

Figure 1.3: Relation between defocalization and trajectory length. Trajectories from RARA-HT tracking experiments were grouped into five bins according to the maximum likelihood estimate for their diffusion coefficient, as indicated on the *x*-axis of the plot. The mean number of jumps across all trajectories in each bin were plotted as points. For these experiments, we measured retinoic acid receptor α fused to HaloTag and tracked with 7.48 ms frame intervals in human U2OS osteosarcoma nuclei. Each point represents a separate biological replicate, and the bar heights are the mean across biological replicates.

Challenge 3: Tracking

Many spaSPT pipelines are PALM/STORM pipelines that have been retrofitted with an additional step - connecting detections into trajectories. Seldom is this a good option. Most of these pipelines have inherent filters against fits that fail to converge (for instance, by imposing limits on the Hessian determinant after a run of Gauss-Newton). Since - due to motion blur - these problems are more likely to occur for fast-moving than slow-moving emitters, they impose inherent state biases beyond those considered in the previous sections. Indeed, it may be preferable to *overdetect*, then identify subsets of detections that are more likely to originate from true trajectories. A localization method with modest accuracy and a 0.1% fail rate may be better than another method with high accuracy and a 5% fail rate.

When data is extremely sparse so that there is fewer than one molecule on average per frame, many tracking algorithms perform comparably. It is in situations with ambiguity - for example, when several particles are in close proximity - that most errors arise. In such situations, the tracking algorithm must make a choice:

1. attempt to resolve the situation, for example by choosing the maximum likelihood set of connections according to some diffusion model;

2. do not attempt to make any connections, discarding this data

Different algorithms handle this choice differently. In a competition that evaluated the accuracy of a variety of tracking methods on the same data [34], no single technique performed the best in all circumstances. While approach (2) seems safer, it may introduce additional biases into spaSPT data. In contrast, when applied naively, approach (1) can return nonsense.

In chapter 2, we examine a simple framework for detection and tracking, descended conceptually from [112], that we have found useful to control the information presented to the tracking algorithm. This framework actually encompasses a variety of detection and tracking algorithms. It can be considered on its own or as a complement to the GitHub repository quot, which also provides a graphic user interface for exploring the use of different detection and tracking algorithms on user datasets.

Challenge 4: Model selection

spaSPT generates thousands of trajectories per cell, each of which may only be a few frames in length. Unlike that of other microscopic modalities, this kind of data is not readily intepretable by a human scientist in its raw state. Instead, interpretation is often performed through the lens of stochastic *diffusion models*. Such approaches boil down thousands of trajectories to a small number of model parameters.

While the use of diffusion models is ubiquitous in spaSPT studies, there are only a few ways to identify an appropriate diffusion model for a given dataset. Among the most important question are:

- How many types of motion ("states") are present in the dataset?
- What kind of motion? (regular Brownian, fractional Brownian, Levy flights, etc.)

For some inference frameworks - for example, when using radial jump histogram fitting to analyze data - using an inappropriate model for a given dataset can have disastrous consequences, as investigated in chapter 4. At the same time, because situations in real biological settings are invariably more complex than diffusion models, any diffusion model is necessarily a simplification. Methods from Bayesian statistics are a promising approach for model selection that balances a model's sparsity against its likelihood given the data in a statistically principled way [50]. An investigation of approaches using these techniques forms the heart of chapter 5. Still, these methods are currently only applicable to a fairly small

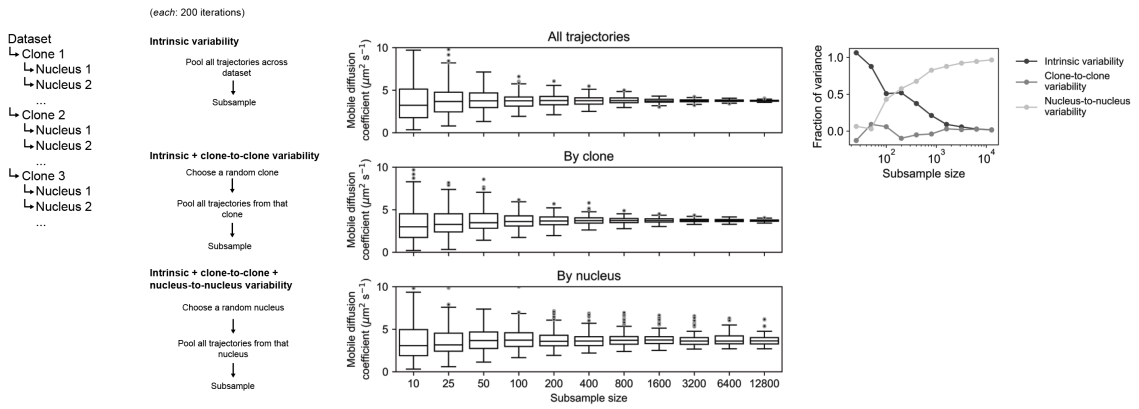

Figure 1.4: Analysis of the origins of variability in an experiment spaSPT dataset. Trajectories are from endogenously tagged retinoic acid receptor α-HaloTag in U2OS nuclei, labeled with the photoactivatable dye PA-JFX549 [15]. Twelve nuclei each from three independent knock-in clones were used for the analysis. The majority of variability at high subsampling counts comes from nucleus-to-nucleus variability.

number of tractable diffusion models. An important area for future work is to develop efficient numerical methods for model inference with a broader category of diffusion models, perhaps using approaches based on Gibbs sampling similar to those outlined in chapter 4 and chapter 5.

Challenge 5: Biological variability

The primary source of noise in spaSPT is cell-to-cell variability ("extrinsic variability"), rather than variability arising from the finite samples of trajectories in any given cell ("intrinsic variability") (Fig. 1.4). The cell cycle, genetic heterogeneity, and cell state heterogeneity mean that spaSPT experiments performed on different cells in a population do not necessarily represent "draws from the same distribution". This poses additional challenges for the interpretation of spaSPT data. For instance, if we identify a distinct subpopulation of trajectories in an spaSPT dataset according to some criterion, it is important to determine whether the subpopulation is found in all cells or is restricted to one or a few cells. Indeed, it may also be important to determine whether the subpopulation is only found in spatially distinct *parts* of a cell.

The best tools for assessing biological variability are bootstrapping and visualization. In particular, in section 3.2 we explore "aggregate likelihood" approaches that are useful for spaSPT practitioners who wish to assess the cell-to-cell and position-to-position variability in their spaSPT dataset.

1.0.1 Outline

The five problems outlined above - localization error, observation geometry, tracking, model selection, and biological variability - have direct consequences for the interpretation of spaSPT data with diffusion models. In this thesis, we derive inference frameworks for diffusion models that are conscious of these constraints. While no single method provides a one-size-fits-all approach to spaSPT analysis, we find that together the techniques constitute a useful toolkit for the spaSPT experimentalist.

In chapter 2, we describe a simple framework for detection and tracking in spaSPT data. The framework actually encompasses a variety of different tracking algorithms, and while unsophisticated, it provides the user with intuitive ways to control the tracking algorithms' behavior in situations with ambiguity. The chapter should be considered a complement to the GitHub repository quot, which implements the algorithms described and also provides a graphic user interface for examining the output of the tracking algorithm along with various other visualization utilities geared toward spaSPT data.

In section 3.1, we review methods to extract parameters governing single-state diffusion models from experimental spaSPT datasets. In particular, we examine three types of diffusion models:

1. regular Brownian motion
2. fractional Brownan motion
3. Levy flights

The first arises as a special case of the second and third. In addition to directly enabling the measurement of parameters for these types of motion in biological settings, the chapter also lays the mathematical groundwork for the combinations of diffusive states considered in later chapters.

In section 3.2, we describe simple nonparametric analyses of spaSPT data. The methods outlined in this chapter can be considered alternatives to the model-based analyses elsewhere in the thesis. However, they are still dependent on constraints in the spaSPT experiment and these dependencies are examined in detail.

In chapter 4, we examine multi-state diffusion models ("mixture models"). A common way to extract parameters for mixture models is radial jump fitting [60], which is reviewed here. In addition, we describe two alternative frameworks for model inference - expectation-maximization and Gibbs sampling - that perform comparably and provide the basis for the methods outlined in the next

chapter. These two alternative frameworks have publicly available implementations at github.com/alecheckert/emdiff (`emdiff`) and github.com/alecheckert/gibberdiff (`gibberdiff`), respectively.

chapter 5 is the heart of the thesis. We attempt to address the issue of model selection in spaSPT data, describing three methods (finite-state variational Bayes, arrayed state samplers, and Dirichlet process mixture models) that perform model selection in combination with parameter inference. We examine the efficiency of these approaches in various simulated and real datasets. The methods described in this section are available as simple, easy-to-use tools in the following software packages:

- github.com/alecheckert/emdiff (`vbdiff`): discrete-state variational Bayes
- github.com/alecheckert/dpsp (`dpsp`): Dirichlet process mixture models

Finally, chapter 6 applies the methods developed in previous chapters to some biological problems of interest. In particular, we focus on how the combination of information from different methods can help provide clarity on the behavior of a particular protein target.

An important subject for future work is to integrate the techniques here with additional dimensions of information accessible to the spaSPT user - for instance, spectral information [35] or the axial dimension [33].

Chapter 2

Detection and tracking algorithms for live cell stroboscopic PALM

Fluorescent single particle tracking (spaSPT) produces a time-indexed sequence of images with spots corresponding to the paths of individual particles convolved with the microscope point spread function (PSF). Prerequisite to any analysis of diffusion is the accurate identification and tracking of individual spots between frames.

Perhaps due to their history as retrofitted PALM/STORM localization pipelines, most spaSPT pipelines operate in three steps:

1. *Particle detection*. The approximate (nearest-pixel) position of spots are identified in each frame.
2. *Subpixel localization*. Identified spots are subjected to an estimation of the subpixel position - for instance, using iterative fitting methods.
3. *Tracking*. Once localized, spots are tracked between frames according to heuristic or probabilistic criteria.

Because camera integration times in spaSPT are often a small fraction of those used in fixed-cell PALM/STORM, the number of photons per spot is often around $\sim$100-300 rather than in the 1000s. In some cases, detected spots may have fewer than 100 photons. In these low-light conditions, it is far more critical to have robust detection and tracking methods than extremely precise localization methods.

Since the ideal detection and tracking methods for spaSPT are not yet known, user supervision to maintain the quality of trajectories is still an important step. To address this need, we produced a combined API and graphic user interface for spaSPT analysis in Python (Fig. 2.1), accessible at quot. This graphic user interface incorporates several sub-GUIs that address different parts of the spaSPT

pipeline, from the initial optimization of detection and tracking settings to review of processed trajectories, and finally to downstream analyses and sub-ROI masking (Fig. 2.1B). Any of the settings accessible in the GUI can also be arranged into pipelines that can be executed in parallel with the `dask` Python library (Fig. 2.1A).

The purpose of this chapter is to provide a reference for the methods in the quot repository. As more detection, localization, and tracking methods are added to quot, this reference will grow. Since the thesis is a static document, users are recommended to quot for an updated set of methods.

2.1 Spot detection

`quot` features a set of basic computer vision methods for particle detection. Many of these are variants of generalized log likelihood ratio tests (GLLRTs), which are explored in detail in Appendix A. GLLRTs are attractive in that they are invariant with respect to the absolute intensities of the underlying image due to implicit normalization against local noise, and so the same detection settings can often be used for spaSPT movies from different cell lines or even on different microscopes.

The GLLRT's property of intensity invariance can also be transferred in a limited form to other detection methods (Appendix A, section A.3). Essentially, this process equips a detection method with a final step prior to thresholding that renormalizes each pixel against local variance, and can be deployed quickly with the FFT. In this way, we created Hessian determinant-based spot detection algorithms that are more intensity-invariant than previous methods (`hess_det`, `hess_det_broad_var` in `quot`).

Other detection methods have been described in detail elsewhere [34].

2.2 Spot localization

Due to motion blur, subpixel localization takes a different form for spaSPT than for fixed-cell PALM/STORM. Specifically, robustness is far more critical than precision. Whereas in fixed-cell PALM/STORM individual detections can always be discarded if localization fails to converge, in spaSPT discarding localizations can pose a problem for downstream tracking analysis.

While a comprehensive analysis of the efficacy of localization methods on PSFs with motion blur is still lacking, we find that often the simplest localization meth-

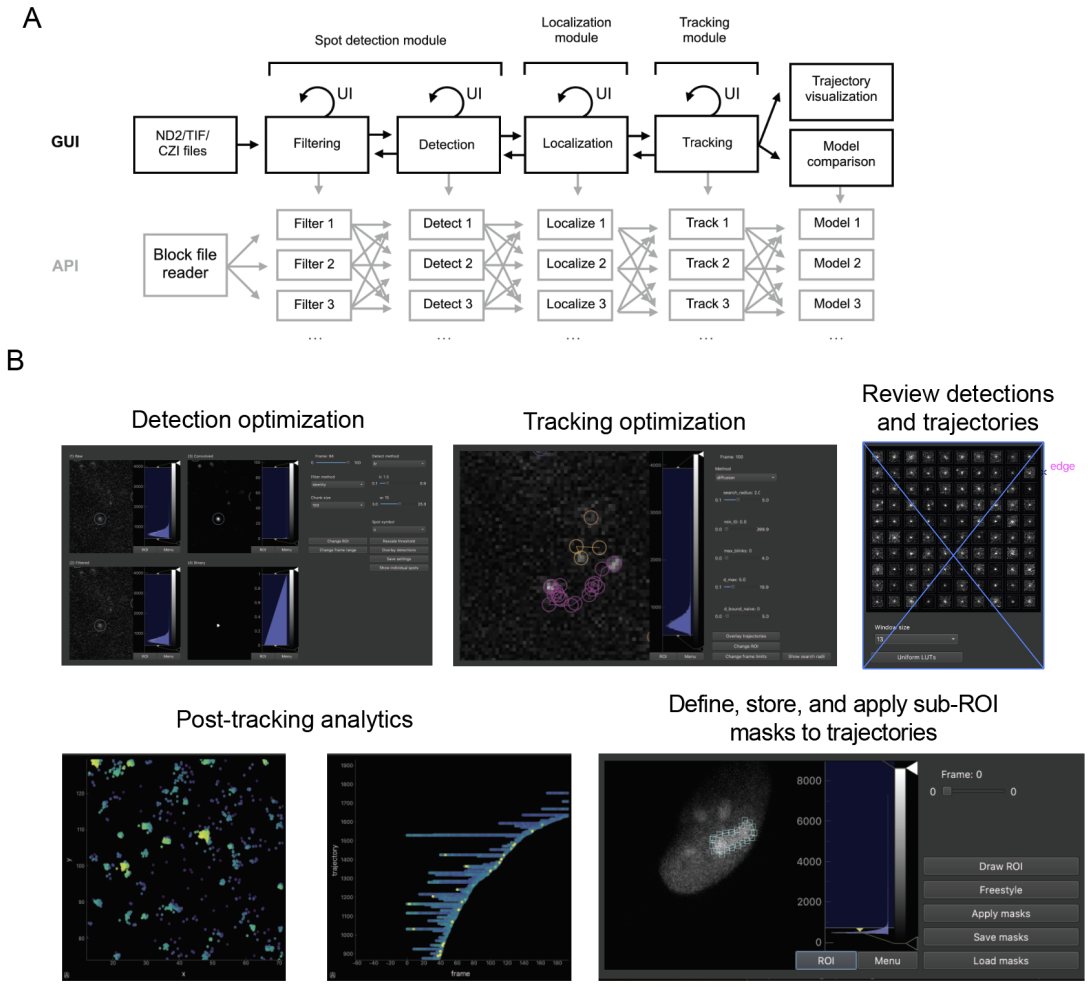

Figure 2.1: Schematic of the quot tool. (A) Steps in typical spaSPT pipelines. "UI" arrows indicate points where iterative user feedback is usually required to improve the method. quot has both a GUI for investigation of different filtering, detection, localization, and tracking methods as well as an API for combining these methods into custom pipelines. (B) Screenshots of sub-GUIs in the quot tool.

ods work best.

Radial symmetry. Proposed by Parthasarathy [39], the radial symmetry method is a non-iterative, non-PSF model based method. Given a spot, the least-squares solution to the point of maximal radial symmetry can be solved in a single step. While the resulting estimator does not perform as well on iterative methods when the PSF model is known, it is highly robust to localization error. In all of the iterative methods in `quot`, radial symmetry is the initial guess used to seed the fit.

Radial symmetry has some pixel center bias (Fig. 2.3A).

Gauss-Newton with pointwise 2D Gaussian PSF. The Gauss-Newton algorithm is an iterative least-squares method to find the maximum likelihood estimator for a model with Gaussian-distributed noise.

Specifically, suppose that $f(x, y \mid \theta)$ is the PSF model evaluated on pixel (x, y), that θ is the vector of model parameters for the PSF, and that $\mathbf{X}$ is the vector of observed pixel intensities so that X_k is the pixel intensity for the pixel at (x_k, y_k). Then the log likelihood of θ given $\mathbf{X}$ under Gaussian noise with variance ν^2 is

$$\log \mathcal{L}[\theta \mid \mathbf{X}] = -\frac{1}{2\nu^2} \sum_{k=1}^{K} (X_k - f(x_k, y_k \mid \theta))^2 - K \log(2\pi\nu^2)$$

This has the gradient

$$\frac{\partial \log \mathcal{L}}{\partial \theta_j} = \frac{1}{\nu^2} \sum_{k=1}^{K} (X_k - f(x_k, y_k \mid \theta) \frac{\partial f}{\partial \theta_j}$$

and the Hessian

$$H_{ij} = \frac{\partial^2 \log \mathcal{L}}{\partial \theta_i \partial \theta_j} = \frac{1}{\nu^2} \sum_{k=1}^{K} (X_k - f(x_k, y_k \mid \theta)) \frac{\partial^2 f}{\partial \theta_i \partial \theta_j} - \frac{1}{\nu^2} \sum_{k=1}^{K} \frac{\partial f}{\partial \theta_i} \frac{\partial f}{\partial \theta_j}$$

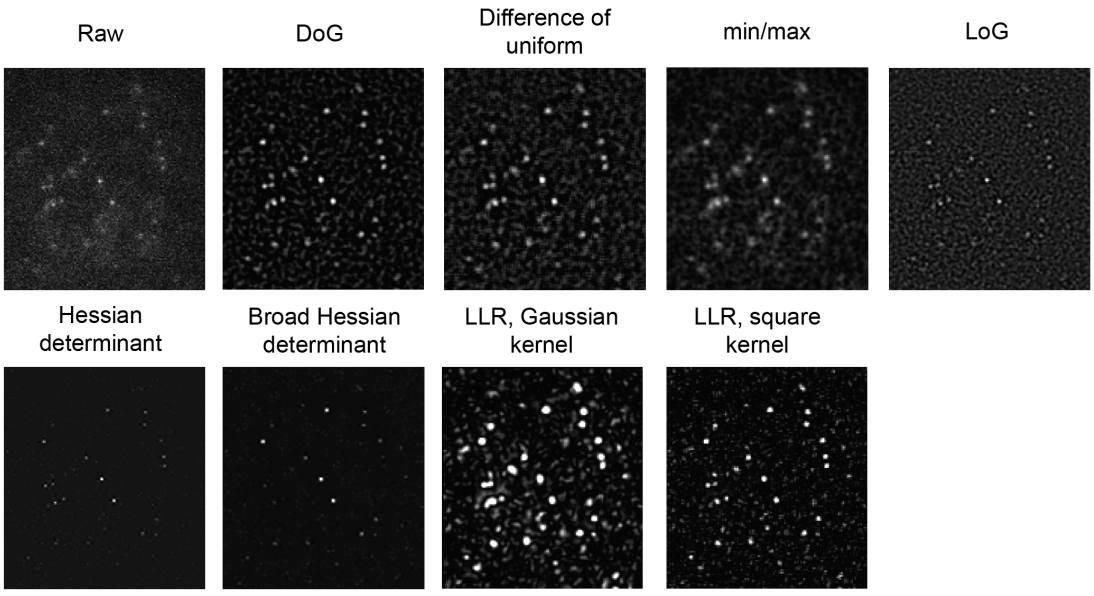

Figure 2.2: Some detections method in the quot package. Both the raw convolved images and post-threshold detections are accessible in the quot interface.

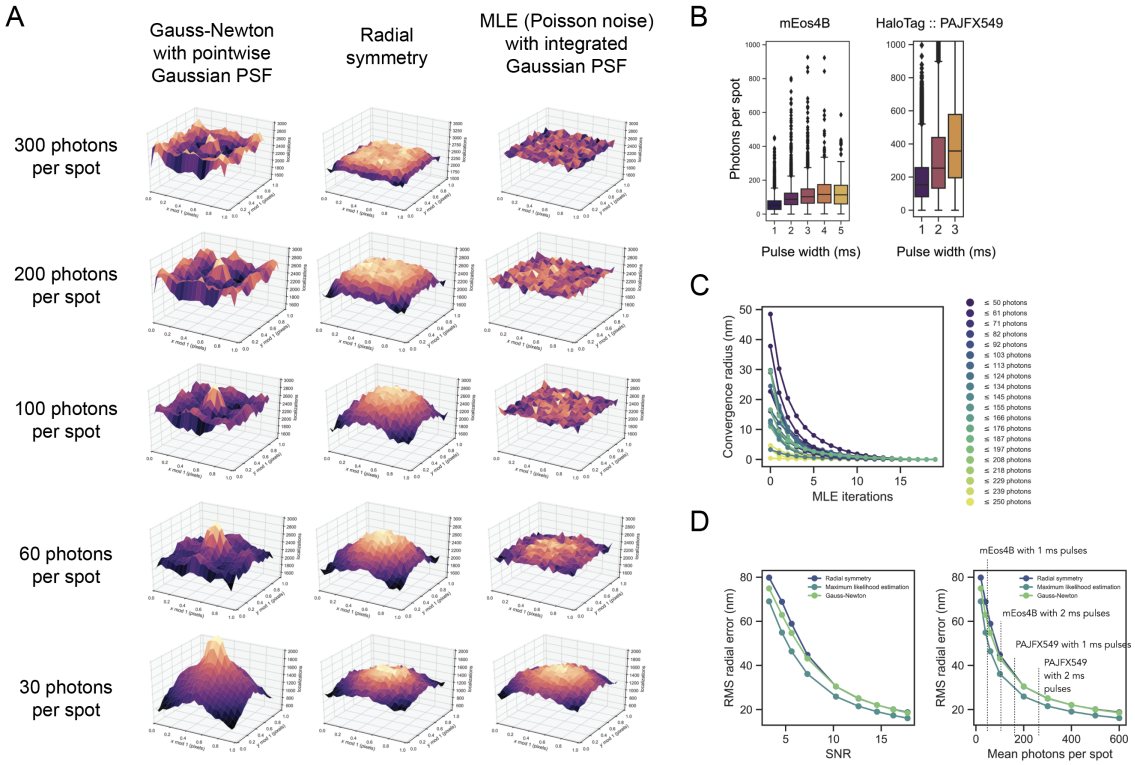

Figure 2.3: Assessing center-edge bias of localization methods on simulated spots in low-light regimes. (A) Moving particles with a Gaussian PSF and radius 200 nm were simulated on a two-dimensional surface and sampled as a Poisson process, then localized with three different algorithms. The position of the localized spot relative to the true spot was plotted as a heat map. (B) Number of photons per spot in real spaSPT data with transfected mEos4B or HaloTag labeled with HTL-PAJFX549. (C) Number of iterations of the "poisson_int_gaussian" algorithm to converge on experimental spaSPT spots. (D) Root mean square radial error of the three localization methods in (A) on simulated spots, with approximate photon regimes.

If we assume that $\partial^2 f / \partial\theta_i \partial\theta_j \approx 0$, then the Hessian can be expressed

$$H_{ij} = -\frac{1}{\nu^2}\mathbf{J}^T\mathbf{J}$$

where $\mathbf{J}$ is the Jacobian with elements

$$J_{ki} = \frac{\partial f(x_k, y_k \mid \theta)}{\partial \theta_i}$$

The multivariate Newton's method for a function g is

$$\theta_{t+1} = \theta_t - \left(\nabla\nabla^T g\right)^{-1} \nabla g$$

Substituting the log likelihood for g, we have

$$\theta_{t+1} = \theta_j + \left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{J}^T\mathbf{r}$$

where $\mathbf{r}$ is the vector of model residuals, so that $r_k = X_k - f(x_k, y_k \mid \theta_t)$.

Because this involves a matrix inversion, we regularize the problem by subtracting a ridge term from the approximate Hessian to guarantee it is negative definite, and then damp the iteration by some $\gamma \in [0, 1]$ so that the final Gauss-Newton algorithm is

$$\theta_{t+1} = \theta_j + \gamma\left(\mathbf{J}^T\mathbf{J} - a\mathbf{I}\right)^{-1}\mathbf{J}^T\mathbf{r}$$

To determine the magnitude of the ridge term a, we use Sylvester's law of inertia, which states that the pivots of the matrix $\mathbf{J}^T\mathbf{J}$ in *LU* form can be used to determine the definiteness of the matrix.

The `ls_point_gaussian` method is obtained by simply letting

$$f(x, y \mid \theta) = \frac{\theta_I}{2\pi\sigma_0^2}\exp\left(-\frac{(x-\theta_x)^2 + (y-\theta_y)^2}{2\sigma_0^2}\right) + \theta_{\text{bg}}$$

for some suitably chosen PSF radius σ_0^2. Four parameters are estimated through Gauss-Newton: the PSF intensity θ_I, the background intensity per pixel θ_{bg}, and the center coordinates of the spot θ_y, θ_x.

Gauss-Newton with integrated 2D Gaussian PSF. The pointwise 2D Gaussian PSF assumes that the intensity across an entire pixel is described by the PSF evaluated in the pixel's center. Because this is only approximately true, it leads to edge-center bias in low-light regimes (Fig. 2.3A).

A solution to the problem can be found by integrated the Gaussian PSF across the borders of each pixel [40]. If we take $\text{PSF}(u, v \mid \theta)$ to be the pointwise evaluated Gaussian PSF from the previous section, then the integrated intensity on the unit-size pixel centered at (x, y) has intensity

$$\begin{aligned} f(x, y \mid \theta) &= \int_{x-\frac{1}{2}}^{x+\frac{1}{2}}\int_{y-\frac{1}{2}}^{y+\frac{1}{2}} \text{PSF}(u, v)\,\mathrm{d}u\,\mathrm{d}v \\ &= \frac{\theta_I}{4}\left[\text{erf}\left(\frac{x-\theta_x+\frac{1}{2}}{\sqrt{2\sigma_0^2}}\right) - \text{erf}\left(\frac{x-\theta_x-\frac{1}{2}}{\sqrt{2\sigma_0^2}}\right)\right] \\ &\quad \cdot \left[\text{erf}\left(\frac{y-\theta_y+\frac{1}{2}}{\sqrt{2\sigma_0^2}}\right) - \text{erf}\left(\frac{y-\theta_y-\frac{1}{2}}{\sqrt{2\sigma_0^2}}\right)\right] + \theta_{\text{bg}} \end{aligned}$$

The rest of the Gauss-Newton algorithm proceeds as before.

Poisson noise MLE with integrated 2D Gaussian PSF. Since shot noise on EMCCD cameras is Poisson-distributed rather than Gaussian-distributed, a more accurate representation of the observed PSFs can be obtained by replacing the Gaussian likelihood function with a Poisson log likelihood function. If $f(x, y \mid \theta)$ is the PSF model for pixel (x, y) and $\mathbf{X}$ is a vector of observed pixel intensities, then this likelihood is

$$\log \mathcal{L}\left[\mathbf{X} \mid \theta\right] = X_k \log f(\theta) - f(\theta) - \log X_k!$$

This corresponds to the gradient

$$\frac{\partial \log \mathcal{L}}{\partial \theta_j} = \frac{\partial f}{\partial \theta_j}\left(\frac{X_k}{f(\theta)} - 1\right)$$

and the Hessian

$$\frac{\partial^2 \log \mathcal{L}}{\partial \theta_j \partial \theta_i} = \frac{\partial^2 f}{\partial \theta_j \partial \theta_i}\left(\frac{X_k}{f(\theta)} - 1\right) - \frac{X_k}{f(\theta)^2}\frac{\partial f}{\partial \theta_j}\frac{\partial f}{\partial \theta_k}$$

Again assuming that $\partial^2 f / \partial \theta_i \partial \theta_j$ can be neglected, we obtain the approximate Hessian

$$H_{ij} = \frac{\partial^2 \log \mathcal{L}}{\partial \theta_j \partial \theta_i} \approx -\frac{X_k}{f(\theta)^2}\frac{\partial f}{\partial \theta_j}\frac{\partial f}{\partial \theta_k}$$

This can then be used in the iterative Levenberg-Marquardt iterative scheme

$$\theta_{t+1} \approx \theta_t - \gamma(\mathbf{H} - a\mathbf{I})^{-1}\nabla \log \mathcal{L}\left[\mathbf{X} \mid \theta\right]$$

where $\gamma \in [0, 1]$ is a damping term and a is a suitably chosen regularization coefficient. This algorithm is equivalent to a Levenberg-Marquardt method with Poisson deviates [41].

Centroid. The centroid algorithm is among the oldest subpixel localization methods, which simplest relies on finding the center of mass of a spot.

2.3 Measuring localization error

A critical step for downstream analysis is determination of the localization error in the experiment. As discussed in the introduction, the localization error is itself a function of the diffusion coefficient [31]. Jointly estimating motion and localization from trajectories is an outstanding problem in spaSPT. In this thesis, we make the

simplifying assumption that localization error is the same for all molecules. Future work must remove this assumption.

Methods to measure localization error for spaSPT data generally fall into one of five categories:

1. Measure the standard deviation of the estimated position of immobile beads.
2. Measure the standard deviation of the estimated position of an immobile molecule inside cells, such as histone H2B-HaloTag. The labeling method for H2B should be identical to the labeling method used on the intended target (photoactivatable fluorophores, photoconvertible proteins, etc.).
3. Measure the mean squared displacement (MSD) of a moving molecule, and extrapolate to the *y*-intercept. This is equal to $2\sigma_{\text{loc}}^2$.
4. Measure the jump length distribution of a molecule with an immobile component (for instance, H2B-HaloTag). Fit the jump length distribution to a model with an immobile state parametrized by the localization error.
5. Invert the so-called observed information matrix (negative Hessian) for the localization problem. The diagonal gives the estimated error for each parameter in localization.

Method (1) is misleading, given that we will obtain far more photons from a stationary bead than from a typical fluorophore. As a result, this method systematically underestimates the localization error associated with real molecules, mobile and immobile alike.

Method (2) is useful, but because it implicitly involves selecting molecules that are immobile, it also systematically underestimates the localization error for moving molecules.

Method (3) is attractive but is sensitive to the same problems as method (2). If immobile trajectories are included in the calculation of the MSD, the localization error for moving particles will be systematically underestimated. A better approach is to stratify the trajectories into populations with distinct estimated diffusion coefficients and take the MSD and localization error of each population separately. So far as we know, this has not been explored.

Method (4) is subject to the same biases as method (2) - it involves calculating the localization error specifically for the immobile population.

Method (5) is dependent on the PSF model used to evaluate the Hessian matrix. While it is a useful way to judge the relative error in different datasets that have

been analyzed with the same localization pipeline, it cannot provide absolute localization error.

We propose an alternative to these methods. If we can measure the diffusion of a molecule with true Markov dynamics (for instance, a purified protein or labeled oligo in solution), then we can exploit the dependence of equation 3.32 (discussed later) on the localization error. The important part here is that the covariance between subsequent 1D jumps in a trajectory is $-\sigma_{\text{loc}}^2$ for a Markov process, independent of time scaling effects. This provides a straightforward way to determine localization error for moving molecules with any diffusion coefficient. The nonzero covariance arises from the mutual dependence of the first and second jumps on the shared localization error inherent in their shared middle point.

Of course, as we will seee in equation 3.31, memory effects such as subdiffusion also contribute to this covariance. Since memory effects are operative in biological diffusion, the ideal case would be to use *in vitro* experiments for determination of the relation between the diffusion coefficient and the localization error.

2.4 Tracking algorithms

Tracking refers to the method by which detections ("spots") are joined to reconstruct trajectories. The goal of tracking is to make *connections* between detections, which are statements of the belief that two detections originate from the same emitter. Tracking algorithms are somewhat less familiar to PALM/STORM practitioners than detection and localization methods, so we focus more attention on them in this chapter.

Intuitively, a trajectory can only be in one place at one time, and - provided we use a fast enough frame interval - trajectories won't move far from one frame to the next. The challenge of tracking, especially in 3D settings, comes from the ambiguity induced by the following effects:

1. If two detections in frame t are both near a detection in frame $t+1$, it is not clear which pair of detections should be connected. The problem becomes more complicated the more detections are in close proximity.

2. Emitters may bleach, defocalize, or blink, causing them to "disappear". In some of these cases (defocalization and blinking), they can reappear at subsequent frames, causing "gaps" in the trajectory.

   Exactly how problematic these gaps are depends on what the user wants to do with the trajectories. Analyses based on hidden Markov models or on

the length of trajectories (binding models) are much more sensitive to the presence of gaps than methods based on jump length distributions.

3. Emitters may photoactivate or enter the focal volume from outside, generating new trajectories.

Here, we outline a set of tracking methods that are useful for spaSPT data. These tracking methods are based around a simple matrix formalism that enables the user to incorporate information from the quality of detections, the local spatiotemporal detection density, and prior beliefs about the nature of the emitters' diffusion.

2.4.1 The detection/tracking problem

The problems of spot detection and connection are intimately connected. Exactly how information from one problem is used to solve the other - that is, how we use the set of observed detections to construct trajectories, or use potential trajectories to decide which detections to record - is one of the primary determinants of spaSPT quality. (It may actually be *the* primary determinant.)

How can we treat this problem? Imagine that we are considering a particular spaSPT movie. Let $\mathbf{S}$ be our set of detections and $\mathbf{R}$ be the set of connections between detections, so that $R_{ij} = 1$ when detections i and j originate from the same emitter and $R_{ij} = 0$ otherwise. The goal of the image processing/tracking steps in spaSPT is to determine $\mathbf{S}$ and $\mathbf{R}$.

There are some natural constraints on $\mathbf{R}$. For instance, we cannot make a connection between detections i and j if they originate from the same frame. For a given $\mathbf{S}$, however, the number of possibilities for $\mathbf{R}$ is still usually very large. As a result, tracking algorithms introduce additional assumptions that reduce the complexity of the problem while introducing a tolerably low amount of bias.

It is useful to categorize detection/tracking algorithms into two classes by exactly how they treat the interaction between $\mathbf{S}$ and $\mathbf{R}$:

1. A detection/tracking algorithm can work with the full joint distribution $p_{\mathbf{S},\mathbf{R}}(\mathbf{s}, \mathbf{r})$. This is the more general class of algorithms. It explicitly acknowledges the relationship between detection and tracking. For instance,

    - We may be less likely to make a connection between spots if our confidence in the detection of those spots is low.
    - Conversely, we may be less likely to detect a spot if the possible connections induced by it have low likelihood.

Because of the high complexity of $p_{\mathbf{S},\mathbf{R}}(\mathbf{s},\mathbf{r})$, additional assumptions are typically required to make the problem tractable. Examples include joint probabilistic data association filters (JPDAFs), applied in radar and sonar [44].

2. Factorizing $p_{\mathbf{S},\mathbf{R}}(\mathbf{s},\mathbf{r}) = p_{\mathbf{R}|\mathbf{S}}(\mathbf{r}|\mathbf{s})p_{\mathbf{S}}(\mathbf{s})$, a detection/tracking algorithm can first determine some likely set of detections $\mathbf{S}'$ using the marginal distribution $p_{\mathbf{S}}(\mathbf{s})$. Then the connections $\mathbf{R}$ are determined using the conditional density $p_{\mathbf{R}|\mathbf{S}}(\mathbf{r}|\mathbf{S}')$ separately.

   In other words: first we get a set of detections, and second we treat the set of detections as a constant when considering the possible connections.

Many algorithms in class 1 (for instance, JPDAFs) work by iteratively sampling the conditional densities $p_{\mathbf{R}|\mathbf{S}}(\mathbf{r}|\mathbf{s})$ and $p_{\mathbf{S}|\mathbf{R}}(\mathbf{s}|\mathbf{r})$ with Markov chains - that is, using Gibbs sampling [45]. As a result, these share the general shortcoming of MCMC methods in that they are limited in the size of the tracking problems that can be treated.

Another major difficulty with algorithms in class 1 is choosing what the form of $p_{\mathbf{S},\mathbf{R}}(\mathbf{s},\mathbf{r})$ should be. This typically involves parametrizing the problem according to prior beliefs.

To see this, suppose are using a Gibbs sampling approach to evaluate $p_{\mathbf{S},\mathbf{R}}(\mathbf{s},\mathbf{r})$, which requires that we model the conditional density $p_{\mathbf{R}|\mathbf{S}}(\mathbf{r}|\mathbf{s})$. The probability of a given connection R_{ij} is a function not only of the number and positions of the detections (which we suppose are contained in $\mathbf{S}$), but also of the mode of their diffusion (Brownian or non-Brownian), whether convection is present, whether each emitter has a distinct mobility, and the error associated with their positions. Unless we are prepared to incorporate all of these variables into the joint distribution, then we run the risk of strongly biasing our results to our previous beliefs about the way our emitters behave. Assumptions about the nature of the emitters' diffusion is no substitute for sparsity.

For both of these reasons, here we only deal with algorithms in class 2. In other words, we'll assume that we already know the set of detections $\mathbf{S}$ and our sole problem is to determine $\mathbf{R}$.

2.4.2 Matrix formalism for connection

Let n_t be the number of detections in frame $t \in \{0, 1, ..., T-1\}$ for a movie with T total frames. Our goal is to determine which, if any, detections in frame t should be connected to detections in frame $t+1$.

Connection matrices

Define the connection matrix $\mathbf{L}$ with size (n_t, n_{t+1}) such that, for any two detections $i \in \{1, ..., n_t\}$ and $j \in \{1, ..., n_{t+1}\}$, element L_{ij} is proportional to the probability that detection i in frame t originates from the same emitter as detection j in frame $t+1$.

In addition, define an "augmented" connection matrix $\mathbf{L}$ with an extra row and column, so its size is $(n_t + 1, n_{t+1} + 1)$. The extra entries represent the probability that a given detection does not connect to *any* detections in the other frame:

- $\tilde{L}_{n_t+1,j}$ is the probability that a localization j in frame $t+1$ does not originate from any emitter present in frame t
- $\tilde{L}_{i,n_{t+1}+1}$ is the probability that a localization i in frame t does not originate from any emitter present in frame $t+1$

For example, if $n_t = 2$ and $n_{t+1} = 3$, then we would have

$$\mathbf{L} = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \end{bmatrix}$$
$$\mathbf{L} = \begin{bmatrix} L_{11} & L_{12} & L_{13} & B_1 \\ L_{21} & L_{22} & L_{23} & B_2 \\ C_1 & C_2 & C_3 & 0 \end{bmatrix} \tag{2.1}$$

$$\mathbf{L}_{400,403} = \mathbf{L}_{400,401}\mathbf{L}_{401,402}\mathbf{L}_{402,403}$$
$$\mathbf{A}_{400,403} = \mathbf{A}_{400,401}\mathbf{A}_{401,402}\mathbf{A}_{402,403}$$

where B_i is the likelihood that detection i in frame t doesn't connect to anything in frame $t+1$, while C_j is the likelihood that detection j in frame $t+1$ doesn't connect to anything in frame t.

If we normalize $\mathbf{L}$ over rows so that $\sum_{j=1}^{n_{t+1}+1} \tilde{L}_{ij} = 1$, then each element of $\mathbf{L}$ can be interpreted as the probability that a trajectory active in frame t connects to each detection in the subsequent frame, or is terminated. Likewise, normalization over columns gives the probability matrix for the same process moving backwards in time. $\mathbf{L}$ will be assumed to be normalized over rows unless otherwise stated.

Because we have defined $L_{ij} = \tilde{L}_{ij}$ for all pairs of detections (i, j), in general the rows of $\mathbf{L}$ will *not* sum to 1. The difference $1 - \sum_j L_{ij}$ is the probability that detection i does not connect to anything in the next frame.

Suppose we have the situation outlined in Fig. 2.4. Let $\mathbf{L}^{(t)}$ be the connection matrix between frames $t-1$ and t, and likewise let $\mathbf{L}^{(t+1)}$ be the connection matrix

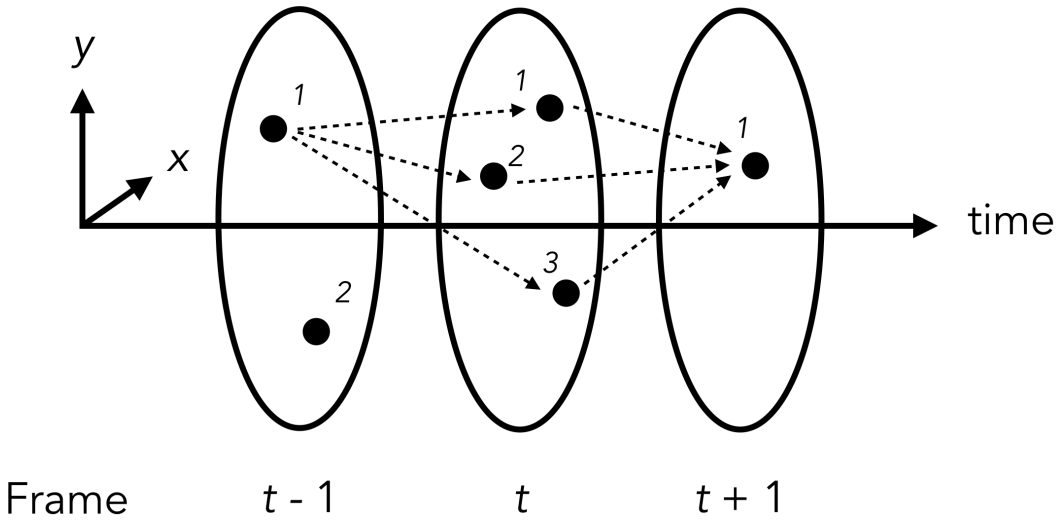

Figure 2.4: A potential situation for connecting detections across subsequent frames. Dots represent detections, which are numbered according to their index in their respective frames. The dotted lines indicate the three possible paths by which detection 1 in frame $t-1$ can connect to detection 1 in frame $t+1$ via an intermediate detection in frame t.

between frames t and $t+1$. Then, to determine how the detections in frame $t-1$ are related to the detections in frame $t+1$, we can form the matrix product

$$\begin{aligned}\mathbf{L}^{(t)}\mathbf{L}^{(t+1)} &= \begin{bmatrix} L_{11}^{(t)} & L_{12}^{(t)} & L_{13}^{(t)} \\ L_{21}^{(t)} & L_{22}^{(t)} & L_{23}^{(t)} \end{bmatrix} \begin{bmatrix} L_{11}^{(t+1)} \\ L_{21}^{(t+1)} \\ L_{31}^{(t+1)} \end{bmatrix} \\ &= \begin{bmatrix} L_{11}^{(t)}L_{11}^{(t+1)} + L_{12}^{(t)}L_{21}^{(t+1)} + L_{13}^{(t)}L_{31}^{(t+1)} \\ L_{21}^{(t)}L_{11}^{(t+1)} + L_{22}^{(t)}L_{21}^{(t+1)} + L_{23}^{(t)}L_{31}^{(t+1)} \end{bmatrix}\end{aligned}$$

Take the first element of this product as an example. This element is the probability that detection 1 in frame $t-1$ connects to detection 1 in frame $t+1$ through an intermediate detection in frame t. Each of the terms represents the probability contributed by one of the three possible routes between these detections, shown by the dotted lines in Fig. 2.4. Because of the possibility that either detection 1 in frame $t-1$ or any of the detections in frame t will be dropped rather than connected, in general $(\mathbf{L}^{(t)}\mathbf{L}^{(t+1)})_{1,1} \neq 1$.

This logic extends to any number of frames. The probability that detection i in frame t_0 will be connected to detection j in frame t_1 through intermediate detec-

tions can be determined by the matrix product

$$\left(\mathbf{L}^{(t_0+1)} \cdots \mathbf{L}^{(t_1)}\right)_{i,j}$$

2.4.3 Search radii and adjacency matrices

In addition to the connection matrix, we can define the *adjacency matrix* $\mathbf{A}$ with shape (n_t, n_{t+1}) such that, for any detections i and j in frames t and $t+1$ respectively,

$$A_{ij} = \begin{cases} 1 & \text{if the distance between } i \text{ and } j \text{ is less than } s_r \\ 0 & \text{otherwise} \end{cases}$$

s_r is the so-called *search radius*, and is defined as an effective upper bound on the displacements of an emitter over a single frame interval. The word "effective" is important here. Suppose we observe a Brownian motion with diffusion coefficient $D = 10\ \mu\text{m}^2\ \text{s}^{-1}$ at 5 ms frame intervals. While there is no upper bound on this particle's displacements, the probability that it diffuses more than 2.0 μm in 2D over the course of a single frame interval is $e^{-(2^2)/(4D\Delta t)} \approx 2 \cdot 10^{-9}$. In order to have a 50-50 chance of observing such a displacement, we would need to measure $\sim 300 \cdot 10^6$ displacements. This is effectively zero for all practical purposes.

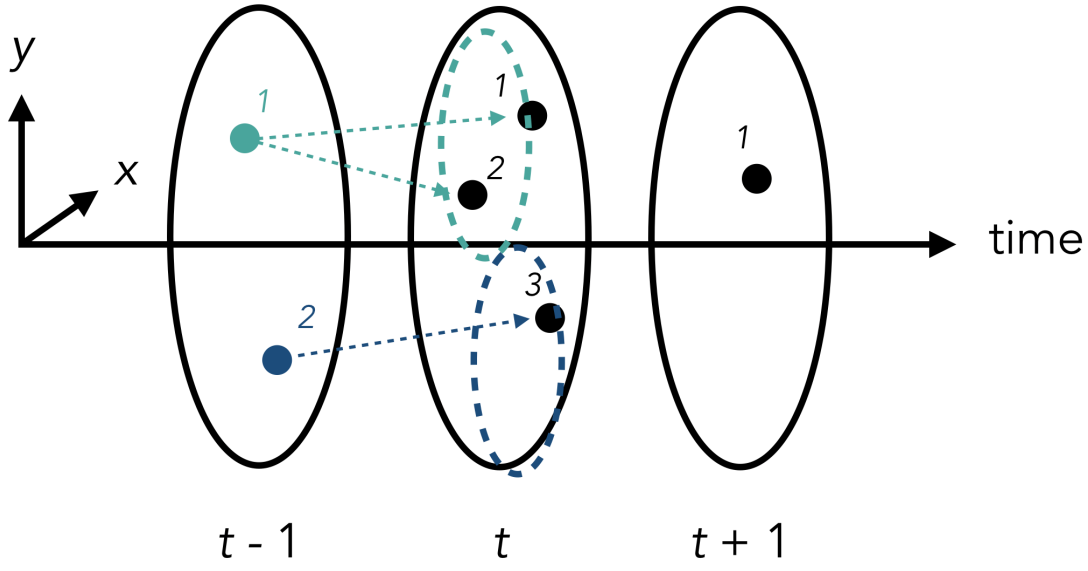

Figure 2.5: Illustration of the role of the search radius. The green and blue circles in frame t represent the search radii of detections 1 and 2 in frame $t-1$, respectively. Dotted-line arrows represent potential reconnections between frame $t-1$ and t.

The search radius has two practical benefits for us:

1. It allows us to reduce the chance of misconnections that arise from a poorly determined connection matrix $\mathbf{L}$. Because the mode of diffusion in a cellular context is never known *a priori*, we can never select $\mathbf{L}$ perfectly and so the search radius fulfills an important regularization function.

2. When chosen intelligently, it can break the connection matrix into separate "subproblems" that are faster to solve without introducing substantial bias into the result.

For instance, suppose we take the example illustrated in Fig. 2.4 and impose a search radius (Fig. 2.5). This produces the adjacency matrices

$$\mathbf{A}^{(t)} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$\mathbf{A}^{(t+1)} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

The first can be broken into two separate subproblems:

$$\mathbf{A}_1^{(t)} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$
$$\mathbf{A}_2^{(t)} = \begin{bmatrix} 1 \end{bmatrix}$$

In the first, we have two detections in frame t competing for a single detection in frame $t-1$. This situation involves ambiguity - we can make an estimate as to which connection to make by assigning weights to each of the edges. In the `euclidean` algorithm in `quot`, these weights are the Euclidean distances between the points, whereas in the `diffusion` algorithm they are the negative log likelihoods for particle (1) in frame $t-1$ to diffuse to either of the two options in frame t, given its past history.

In contrast, $\mathbf{A}_2^{(t)}$ contains a single potential reconnection (assuming our search radius is indeed an upper bound on the jumps). In the `conservative` algorithm in `quot`, this is the only reconnection that we would make; the potential connections in $\mathbf{A}_1^{(t)}$ would be discarded and detections 1 and 2 would be used to seed new trajectories. Notice that while the `conservative` algorithm tends to make higher-confidence connections, it is also biased toward molecules that inhabit less dense parts of the cell.

Finally, there are several categories of tracking errors that can arise from these algorithms:

1. In situations with multiple potential connections (such as $\mathbf{A}_1^{(t)}$ in the example above), we may choose the wrong connections. Since trajectories are unpredictable, even a carefully chosen weighting scheme for the likelihood matrix will go awry in a subset of cases.

2. A trajectory can be connected to a false detection (not a real particle). Since detection with the GLLRT is excellent, we rarely have this problem.

3. A trajectory can be dropped due to a missing detection. This is a far more common error.

4. A trajectory can defocalize or bleach at the same time that another trajectory enters the focal plane or photoactivates.

Of these errors, the last is the most difficult to detect and also surprisingly common in spaSPT data. Note that because the adjacency matrix is 1×1 in these cases, these errors also escape the notice of the `conservative` algorithm. A potential solution to these problems is to impose a limit on the local spatiotemporal density of particles prior to tracking.

2.5 Summary

Detection, localization, and tracking are fundamental prerequisites to model-based analysis of spaSPT data. By providing a tool that can help the user to compare simple spaSPT pipelines, we hope to facilitate the identification of algorithms that work better than existing methods and also to enable users to quickly assess whether existing tracking parameters are transferable to new settings, such as a new cell line or organism.

Chapter 3

Single diffusing states

Fluorescent single particle tracking (spaSPT) enables the measurement of mobility coefficients for single molecular species in complex mixtures, including the intracellular environment. These mobility coefficients - such as the diffusion coefficient for regular Brownian motion - often represent the primary outputs of the spaSPT experiment. As a consequence, the descriptive repertoire of the spaSPT assay is fundamentally limited by the ability to identify and infer model parameters for distinct types of motion in realistic settings.

In this chapter, we examine methods to extract mobility coefficients from spaSPT datasets while accounting for known experimental biases. We derive robust estimators for three types of motion - regular Brownian motion, fractional Brownian motion, and Levy flights. As workhorse models in diffusion modeling, these three provide a simple framework to parametrize categories of molecular behavior of biological interest, including memory (fractional Brownian motion) and balance between local and remote exploration (Levy flights). Together, the methods presented here extend the types of motion measurable with spaSPT and provide the formal basis for multi-state mixture models considered in subsequent chapters.

The chapter is subdivided into two parts:

In the first part, we derive estimators for the three types of motion listed above and evaluate their accuracy on simulated spaSPT datasets. We highlight methods to account for localization error and defocalization in these measurements. Many of the results in this section are used by subsequent chapters.

In the second part, we examine methods to identify the type of motion when this information is not known *a priori*. Classic approaches such as the mean-squared displacement (MSD) method are compared with newer approaches such as angular distributions, covariance matrices, and aggregate likelihood functions. While

many of the approaches are found to be insufficient on their own to deal with the problems posed by localization error and defocalization, together they provide a first-pass toolkit to identify characteristics of motion in spaSPT datasets in the absence of prior information about the mobility of the molecule in question.

3.1 Model-based analysis of single diffusing states

A homogeneous population of diffusing molecules is characterized by some measure of spatial dispersion per unit time - such as the diffusion coefficient. In addition, for non-normal diffusion, there may be one or more "anomaly" parameters that subcategorize the mode of diffusion, such as the Hurst parameter for fractional Brownian motion (FBM) or the stability parameter for Levy flights. The central goal of this part is to relate these model parameters to concrete observables in spaSPT data such as the radial jump histogram.

We begin by reviewing the increment distributions of regular Brownian motion (RBM), focusing on the radial increment (or "jump") distributions as well as the squared jump distributions. We then examine the mean squared displacement (MSD), which is the maximum likelihood estimator for the diffusion coefficient for regular Brownian motion. While the MSD has been a workhorse for historical SPT analysis, we see that localization error places a caveat into most MSD-based analyses. At the end, we discuss least-squares fitting approaches based on the radial jump histogram. This can be seen as an alternative to MSD that extends easily to non-normal diffusion models. Such models, however, do not always share the separability property of RBM's jump distributions. As a result, the finite focal depth of most spaSPT setups has a strong effect on the resulting estimators.

Before continuing, we make one remark concerning the motivation for the way we have structured this chapter. While there are countless models that can be used to interpret spaSPT data, a small number are actually useful for the spaSPT practitioner. In this case, "useful" does not just mean that the model is an accurate representation of the physical and biological process under study, but also that (1) there are simple, nonparametric ways to rule out the model as appropriate to apply to a given dataset, (2) the model is tractable, (3) the model provides information in a form interpretable by humans, and (4) the model is conscious of important experimental constraints in spaSPT (such as localization error and depth of field).

3.1.1 Regular Brownian motion

Definition

We define *regular Brownian motion* as Fickian diffusion with Gaussian-distributed jumps.

Concerned primarily with the movement of gases across fluid membranes in the human circulatory system, the physician-physicist Adolf Fick drew on Joseph Fourier's theory of heat conduction to formulate the first mathematical theory of diffusion in 1855 [2]. If $c(\mathbf{r}, t)$ is the concentration of solute at position $\mathbf{r}$ and time t, then Fick's first law states that the flux of the solute through this surface is proportional to the local concentration gradient

$$\mathbf{J} = -D\nabla c \qquad \text{(first law)}$$

$$\frac{\partial c}{\partial t} = -\nabla \cdot \mathbf{J} \qquad \text{(conservation relation)}$$

$$\frac{\partial c}{\partial t} = \nabla \cdot (D\nabla c)$$

If D is the same everywhere, then the last equation is just $\partial c/\partial t = D\nabla^2 c$, which is known as Fick's second law. (We ignore additional convective terms in this thesis.).

Diffusion processes that obey Fick's first and second laws are called *Fickian*. We highlight the points at which non-Fickian diffusion can arise:

1. The flux is not proportional to the concentration gradient, breaking the first law.
2. There are sources or sinks of solute, violating the conservation relation.
3. The diffusion coefficient varies as a function of position or concentration.

All of these violations occur in biological systems. For instance:

1. Diffusion of a solute in fast binding equilibrium with an immobile scaffold manifests as diffusion with a slower diffusion coefficient [43]. With a fixed number of binding sites per unit volume, the fraction of particles bound to the scaffold decreases with increasing concentration. As a result, the apparent diffusion coefficient becomes concentration-dependent.
2. Barriers to motion - such as large macromolecular complexes - break Fick's first law.
3. Synthesis and degradation of proteins violate the conservation relation.

4. The cell is not homogeneous, varying in composition and viscosity from place to place. Consequently even inert probes that do not interact with any of the cellular components have a diffusion coefficient that is generally a function of position [19].

Fractional Brownian motion and Levy flights provide frameworks to parametrize deviations from Fickian motion.

Under what conditions is regular Brownian motion to be expected in spaSPT experiments? Examining the motion of a particle suspended in a solution at thermal equilibrium, Einstein used an implicit version of the central limit theorem to derive the movements of the particle as Fickian with Gaussian-distributed jumps [36]. While this is mathematically convenient - of course, the Gaussian distribution is the Green's function for Fick's second law - its realm of applicability is restricted to situations where the sources of force on a particle have correlation times far below the measurement interval. In other words, the central limit theorem must hold. The sources of noise in biological experiments, however, rarely satisfy this criterion. They include contributions with very long correlation times, such as the movements of molecular motors, organelle rearrangements, and cell motility. As a result, deviations from regular Brownian motion are routinely observed in a variety of systems, even when the long-term motion is Fickian [79] [37] [38]. This situation has been termed "Brownian, yet non-Gaussian" diffusion. Attempts to provide a model-based framework of this mode of motion that do not incorporate memory effects include a distribution over the diffusivity [79] or a diffusing diffusivity [37].

Alternative definition as a Gaussian process

Regular Brownian can alternatively be described as a Gaussian process X_t with covariance function $\text{Cov}(X_t, X_s) = D \cdot \min(t, s)$. This approach is discussed in detail Appendix B, and is particularly convenient when accounting for localization error.

3.1.2 Jump distributions for regular Brownian motion

The Gaussian character of an RBM's jumps mean that its jump distributions have a simple form. Radial jumps distributions also play a central role in some current frameworks for inferring model parameters in spaSPT data [59] [79] [60].

Suppose that we have an RBM with position $\mathbf{R}$ and that the probability density function for $\mathbf{R}$ is $f_{\mathbf{R}}(\mathbf{r}, t)$. Assuming that concentration and probability are interchangeable, $f_{\mathbf{R}}$ is a solution to Fick's second law. Taking the Fourier transform of the second law,

$$\frac{\partial \phi_{\mathbf{R}}}{\partial t} = -D \left|\mathbf{k}\right|^2 \phi_{\mathbf{R}}$$

where $\phi_{\mathbf{R}}$ is the characteristic function for $\mathbf{R}$, and we have assumed that $f_{\mathbf{R}}$ has a finite support. Integrating from time 0 to t, this has the solution $\phi_{\mathbf{R}}(\mathbf{k}, t) = \phi_{\mathbf{R}}(\mathbf{k}, 0)e^{-D|\mathbf{k}|^2 t}$. If the particle starts out at the origin, then $\phi_{\mathbf{R}}(\mathbf{k}, 0)$ is unity and we have the Green's function

$$\phi_{\mathbf{R}}(\mathbf{k}, t) = e^{-D|\mathbf{k}|^2 t} \tag{3.1}$$

This is the jumping off point for Levy flights, which modify the Green's function by changing the exponent on $|\mathbf{k}|$. We continue to focus on RBMs for now.

3.1 is separable in the frequency coordinate, which implies separability in real space. As a result, 3.1 corresponds to the PDF

$$f_{\mathbf{R}}(\mathbf{r}, t) = \frac{1}{(4\pi D t)^{m/2}} \exp\left(-\frac{|\mathbf{r}|^2}{4Dt}\right)$$

where m is the number of spatial dimensions. In a single dimension,

$$f_X(x, t \mid D) = \frac{1}{\sqrt{4\pi D t}} \exp\left(-\frac{x^2}{4Dt}\right) \tag{3.2}$$

In real spaSPT experiments this density is never encountered, due to the ubiquity of error associated with the estimation of the particle's position. Suppose our error is a random vector $\mathbf{W}$ given by the multivariate normal density

$$\mathbf{W} \sim \mathcal{N}\left(\mathbf{0}, 2\sigma_{\text{loc}}^2 I\right)$$

where I is an $n \times n$ identity matrix and σ_{loc}^2 is the localization error associated with the position of a particle along any single axis. The factor 2 appears because for any observed jump, we have error associated with both the first and the second points that define the endpoints of the jump [55].

Then $\mathbf{W}$ has the CF

$$\phi_{\mathbf{W}}(\mathbf{k}) = \exp\left(-\sigma_{\text{loc}}^2 |\mathbf{k}|^2\right)$$

Define $\overline{\mathbf{R}} = \mathbf{R} + \mathbf{W}$, the random n-dimensional jump of an RBM with localization error. Assuming that $\mathbf{W}$ and $\mathbf{R}$ are independent, then we can apply eq. C.6 from Appendix C to derive the corresponding CF:

$$\begin{aligned}\tilde{f}_{\overline{\mathbf{R}}}(\mathbf{k}, t) &= \phi_{\mathbf{W}}(\mathbf{k})\tilde{f}_{\mathbf{R}}(\mathbf{k}, t \mid D) \\ &= \exp\left(-(Dt + \sigma_{\text{loc}}^2)\,|\mathbf{k}|^2\right)\end{aligned}$$

This corresponds to the PDF

$$f_{\overline{\mathbf{R}}}(\mathbf{r}, t) = \frac{1}{(4\pi(Dt + \sigma_{\text{loc}}^2))^{m/2}} \exp\left(-\frac{|\mathbf{r}|^2}{4(Dt + \sigma_{\text{loc}}^2)}\right) \tag{3.3}$$

Along a single dimension,

$$f_X(x,t) = \frac{1}{\sqrt{4\pi(Dt+\sigma_{\text{loc}}^2)}} \exp\left(-\frac{x^2}{4(Dt+\sigma_{\text{loc}}^2)}\right) \tag{3.4}$$

In subsequent sections, we'll use $\mathbf{R}$ instead of $\overline{\mathbf{R}}$ to denote the position of an RBM with localization error, for simplicity.

Squared 1D jumps of an RBM

The point of the preceding discussion, apart from getting the central equations 3.3 and 3.4, was to stress that the increments of an RBM along each spatial dimension are independent stochastic processes. This property is key to using squared jump-based methods, as we investigate here.

Let X be the jump of an RBM with localization error in one dimension. We've seen that X has the PDF given by eq. 3.4. Let $S = X^2$ be the corresponding squared jump. Then the CDF of S can be written

$$\begin{aligned} F_S(s) &= \Pr(S \le s) = \Pr(X^2 \le s) = \Pr(X \le \sqrt{s}) - \Pr(X \le -\sqrt{s}) \\ &= F_X(\sqrt{s}) - F_X(-\sqrt{s}) \end{aligned}$$

The corresponding PDF is

$$\begin{aligned} f_S(s) &= \frac{\partial F_S}{\partial s} \\ &= \frac{\partial}{\partial s}\left(F_X(\sqrt{s}) - F_X(-\sqrt{s})\right) \\ &= \frac{1}{\sqrt{4\pi s(Dt+\sigma_{\text{loc}}^2)}} \exp\left(-\frac{s}{4(Dt+\sigma_{\text{loc}}^2)}\right), \qquad (s \ge 0) \end{aligned}$$

Comparing with the standard gamma density

$$f_{\text{gamma}}(s \mid \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} s^{\alpha-1} e^{-\beta s} \qquad (s \ge 0) \tag{3.5}$$

we see that

$$S \sim \text{Gamma}\left(\frac{1}{2},\ \frac{1}{4(Dt+\sigma_{\text{loc}}^2)}\right)$$

This gamma density has the characteristic function

$$\phi_S(k) = \left(\frac{1}{1-4ik(Dt+\sigma_{\text{loc}}^2)}\right)^{1/2} \tag{3.6}$$

Squared radial jumps of an RBM in n dimensions

Consider now the squared radial displacement of a random vector $\mathbf{R} = (X_1, ..., X_n)^T$ in m dimensions distributed according to eq. 3.3: $S = X_1^2 + ... + X_m^2$. We've seen that the jumps X_i are mutually independent random variables, and that each X_i^2 has a CF given by eq. 3.6. So, applying the convolution property eq. C.6, their sum has the CF

$$\phi_S(k) = \left(\frac{1}{1 - 4ik(Dt + \sigma_{\text{loc}}^2)} \right)^{n/2} \tag{3.7}$$

This means that S is another gamma random variable with the density

$$S \sim \text{Gamma}\left(\frac{m}{2}, \frac{1}{4(Dt + \sigma_{\text{loc}}^2)} \right) \tag{3.8}$$

Since the expected value for a gamma random variable 3.5 is α/β, we have

$$\mathbb{E}[S] = 2n\left(Dt + \sigma_{\text{loc}}^2\right)$$

which is the familiar MSD for RBM.

Radial displacements of an RBM in m dimensions

Finally, we seek the distribution of the root squared radial displacement $R = \sqrt{S}$. We'll usually refer to R simply as the *radial jump*.

Examining the CDF of R,

$$\begin{aligned} F_R(r) &= \Pr(R \leq r) = \Pr\left(S \leq r^2\right) \\ &= F_S(r^2) \end{aligned}$$

which corresponds to the PDF

$$\begin{aligned} f_R(r) &= \frac{\partial F_S(r^2)}{\partial r} \\ &= 2r\, f_S(r^2) \end{aligned}$$

Using 3.8, this is

$$f_R(r) = \frac{2r^{m-1} \exp\left(-\frac{r^2}{4(Dt + \sigma_{\text{loc}}^2)} \right)}{\Gamma\left(\frac{m}{2}\right) \left(4(Dt + \sigma_{\text{loc}}^2)\right)^{\frac{n}{2}}} \tag{3.9}$$

for $r \geq 0$, and 0 otherwise.

Taking the CDF $F_R(r) = \int_{-\infty}^{r} f_R(r')\mathrm{d}r'$, we find that it has a simple form:

$$F_R(r) = \begin{cases} \gamma_l\left(\frac{m}{2}, \frac{r^2}{4(Dt+\sigma_{\text{loc}}^2)}\right) & \text{if } r \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

where γ_l is the regularized lower incomplete gamma function, defined by

$$\gamma_l(\alpha, x) = \frac{1}{\Gamma(\alpha)} \int_0^x t^{\alpha-1} e^{-t} \mathrm{d}t$$

Another useful metric is the mean radial distance traversed by the RBM after time t:

$$\mathbb{E}[R] = \frac{\Gamma\left(\frac{m+1}{2}\right)}{\Gamma\left(\frac{m}{2}\right)} \sqrt{4(Dt + \sigma_{\text{loc}}^2)}$$

Special cases

The radial displacements of a Brownian motion (eq. 3.9) have well-known special cases for numbers of spatial dimensions m.

If $m = 1$, then we recover the usual 1D jump for an RBM:

$$f_R(r) = \frac{2}{\sqrt{4\pi(Dt + \sigma_{\text{loc}}^2)}} \exp\left(-\frac{r^2}{4(Dt + \sigma_{\text{loc}}^2)}\right)$$

If $m = 2$, we recover the Rayleigh distribution:

$$f_R(r) = \frac{r}{2(Dt + \sigma_{\text{loc}}^2)} \exp\left(-\frac{r^2}{4(Dt + \sigma_{\text{loc}}^2)}\right) \tag{3.11}$$

If $m = 3$, we recover the Maxwell-Boltzmann distribution:

$$f_R(r) = \sqrt{\frac{2}{\pi}} \frac{r^2 \exp\left(-\frac{r^2}{4(Dt+\sigma_{\text{loc}}^2)}\right)}{(2(Dt + \sigma_{\text{loc}}^2))^{\frac{3}{2}}} \tag{3.12}$$

and so on.

The various special cases above raise an interesting question: does the density 3.9 have any physical meaning for noninteger $m \in \mathbb{R}$? Certainly this defines the jump distribution of a valid diffusion process, at least if we drop the localization error term for the moment. If we require that this hypothetical diffusion process is

embedded in a 3D space and has jumps $\mathbf{R}$ whose radial magnitude R is distributed according to 3.9 and whose direction is uniformly selected from the surface of a sphere, then the displacements in the x, y, and z dimensions are distributed according to

$$f_{X,Y,Z}(x,y,z) \propto \left(x^2+y^2+z^2\right)^{\frac{m-1}{2}} \exp\left(-\frac{x^2+y^2+z^2}{4Dt}\right)$$

This density is only separable when $m = 3$, matching the dimension of the space in which the process is embedded. Noting that the characteristic function for the squared radial displacements is still given by eq. 3.7, if we attempt to require that the CF separate into a sum of iid gamma random variables we run into singularities at the origin. This is physically unreasonable, so it is likely that if such a model does describe a diffusion process of physical origin, it cannot arise from separable processes in the x, y, and z dimensions. Diffusion in systems with non-integer geometry has been considered by numerous other authors, most notably by Ben-Avraham and Havlin [57]. We do not comment on these densities further in this thesis.

3.1.3 Maximum likelihood estimator for diffusion coefficient

Given a particular trajectory, how can we infer its diffusion coefficient and how accurately can we do it?

Consider an RBM with jumps $\mathbf{X} = (X_1, ..., X_L)$ and $\mathbf{Y} = (Y_1, ..., Y_L)$ in the x and y directions, respectively. We have L jumps in this trajectory, so the trajectory has a total of $L + 1$ points. As outlined in Appendix B, $\mathbf{X}$ and $\mathbf{Y}$ are independent with

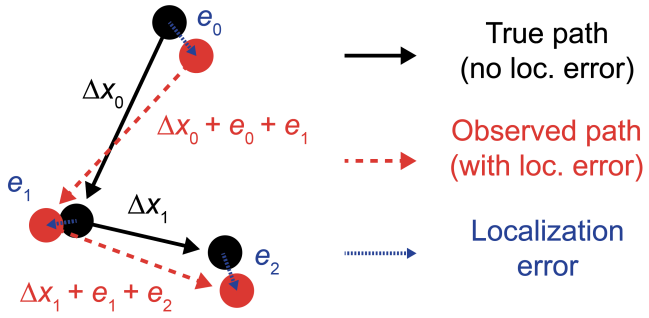

Figure 3.1: Visualization of two sequential jumps in a trajectory, with and without the influence of localization error. Even when the Δx_1 and Δx_2 are independent, the observed jumps are not due to the mutual dependence on the localization error e_1.

probability density functions given by

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{\exp\left(-\frac{1}{2}\mathbf{x}^T C_\Delta^{-1}\mathbf{x}\right)}{(2\pi)^{\frac{1}{2}}\det(C_\Delta)^{\frac{1}{2}}}$$
$$f_{\mathbf{Y}}(\mathbf{y}) = \frac{\exp\left(-\frac{1}{2}\mathbf{y}^T C_\Delta^{-1}\mathbf{y}\right)}{(2\pi)^{\frac{1}{2}}\det(C_\Delta)^{\frac{1}{2}}}$$

where

$$(C_\Delta)_{ij} = 2(D\Delta t + \sigma_{\text{loc}}^2)\mathbb{I}_{i=j} - \sigma_{\text{loc}}^2\mathbb{I}_{|i-j|=1} \tag{3.13}$$

Examining the structure of the covariance matrix C_Δ, notice that it can be viewed as the sum of a ridge and off-diagonal term:

$$C_\Delta = 2(D\Delta t + \sigma_{\text{loc}}^2)\begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix} - \sigma_{\text{loc}}^2\begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & 1 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \end{bmatrix}$$

All of the covariance between the components of $\mathbf{X}$ or $\mathbf{Y}$ - that is, between subsequent jumps in the trajectory - comes from the localization error in the off-diagonal terms. Only when $\sigma_{\text{loc}}^2 = 0$ does the matrix become diagonal. In this ideal case, all of the jumps (X_i, Y_i) are mutually independent.

For the moment we'll assume that $\sigma_{\text{loc}}^2 = 0$ and return to the nonzero case later. In this case, the covariance matrix reduces to $C_\Delta = (2D\Delta t)I$ and its inverse is $(2D\Delta t)^{-1}I$. The determinant is $(2D\Delta t)^L$.

Because $\mathbf{X}$ and $\mathbf{Y}$ are independent, their joint density is just the product of their marginal densities:

$$f_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y}) = \frac{\exp\left[-\frac{1}{2}\left(\mathbf{x}^T C_\Delta^{-1}\mathbf{x} + \mathbf{y}^T C_\Delta^{-1}\mathbf{y}\right)\right]}{2\pi\det(C_\Delta)} \tag{3.14}$$

The corresponding log density is

$$\log f_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y}) = -\frac{1}{2}\left(\mathbf{x}^T C_\Delta^{-1}\mathbf{x} + \mathbf{y}^T C_\Delta^{-1}\mathbf{y}\right) - \log(2\pi) - \log\det(C_\Delta)$$

Substituting the covariance identities for the case $\sigma_{\text{loc}}^2 = 0$, this becomes

$$\log f_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y}) = -\frac{1}{4D\Delta t}\left(\mathbf{x}^T\mathbf{x} + \mathbf{y}^T\mathbf{y}\right) - \log(2\pi) - L\log(2D\Delta t) \tag{3.15}$$

This is solely a function of $\mathbf{x}^T\mathbf{x}+\mathbf{y}^T\mathbf{y}$, the sum of squared displacements. Seek the maximum likelihood estimator $\hat{D}$ by differentiating this density with respect to D and setting the result equal to zero:

$$\frac{\partial \log f_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y})}{\partial D}\Big|_{D=\hat{D}} = \frac{\mathbf{x}^T\mathbf{x}+\mathbf{y}^T\mathbf{y}}{4\hat{D}^2\Delta t} - \frac{L}{\hat{D}} = 0$$

and so we have the MLE

$$\begin{aligned}\hat{D} &= \frac{\sum_{j=1}^{m}\left(x_j^2+y_j^2\right)}{4L\Delta t} \\ &= \frac{1}{4\Delta t}\mathbb{E}\left[X^2+Y^2\right]\end{aligned} \tag{3.16}$$

where X and Y are the x and y components of any displacement in the trajectory. Thus, when localization error is absent, the mean squared displacement is the maximum likelihood estimator for the diffusion coefficient of a single diffusing state.

Influence of localization error on the diffusion coefficient MLE

The reader may notice that there is a much simpler route to deriving the estimator 3.16. We could have simply modeled each jump in the trajectory as an independent draw from the PDF 3.9. The PDF for the sum of these jumps would then be equivalent to equation 3.15.

The reason we built the result from the multivariate normal density 3.14 is that it stresses the rather nonintuitive role that localization error plays in the estimator. To the point: when $\sigma_{\text{loc}}^2 > 0$, then the MSD ceases to be the maximum likelihood estimator for the diffusion coefficient [30] [32]. The effect is illustrated in Fig. 3.1. Two sequential jumps in a trajectory, no matter how many dimensions the trajectory is measured in, will always be codependent due to their mutual dependence on the localization error in their shared point. This is true even when the trajectory is a Markov process, as in the case of RBM.

The magnitude of the covariance between sequential jumps is $-\sigma_{\text{loc}}^2$, and manifests in the off-diagonal terms of the covariance matrix 3.13. This has been investigated in detail in the previous chapter.

Let's attempt, however, to stomach the assumption that sequential jumps in the same trajectory are independent - even in the presence of localization error. In this case, the sum of squared displacements is distributed according to the gamma density

$$S = \sum_{j=1}^{L} \left(Y_j^2 + X_j^2\right) \sim \text{Gamma}\left(\frac{mL}{2}, \frac{1}{4(Dt + \sigma_{\text{loc}}^2)}\right) \tag{3.17}$$

where we have applied the convolution property C.6 to 3.8. Taking the two-dimension case ($m = 2$) and writing the log density explicitly, we have

$$\log f_S(s) = (L-1)\log(s) - \frac{s}{4(D\Delta t + \sigma_{\text{loc}}^2)} - \log\Gamma(L) - L\log(4(D\Delta t + \sigma_{\text{loc}}^2))$$

Differentiating this density with respect to D and solving for the maximum likelihood estimator, we find

$$\hat{D} = \frac{(s/4L) - \sigma_{\text{loc}}^2}{\Delta t} \tag{3.18}$$

which is just a shifted version of 3.16. Notice that this may lead to negative estimates of the diffusion coefficient.

3.1.4 Cramer-Rao lower bound for MSD estimators

Trajectories generated by spaSPT modalities are often very short - each trajectory often has as few as 3 to 4 points. Given this limited information, we may ask: how accurately can we estimate the diffusion coefficient from a single trajectory?

A natural choice to evaluate the accuracy of the MSD estimator is the Cramer-Rao lower bound (CRLB), which we briefly summarize here. Suppose that $f_X(x \mid \theta)$ is the probability density for a random variable X dependent on parameter θ. Imagine that we have a method to generate an estimate of θ based on an observation of X. We'll call this estimate $\hat{\theta}$.

Imagine that we run this estimator many times on independent datasets. We should hope that our estimator guesses θ correctly on average. That is, we should hope that

$$\mathbb{E}_{X|\theta}\left[\hat{\theta} - \theta\right] = 0$$

Here, θ is the true parameter value, $\hat{\theta}$ is our estimate, and the expectation is defined respective to the conditional density of X given θ:

$$\mathbb{E}_{X|\theta}\left[\hat{\theta} - \theta\right] = \int \left(\hat{\theta} - \theta\right) f_{X|\theta}(x|\theta)\mathrm{d}x$$

When this is satisfied, our estimator is *unbiased*. Still, the estimator may not be very good. If it tends *on average* to guess the parameter correctly, but with a high variance, it might still be useless. As such, we should seek an estimator that also

tends to return parameters close to the real parameter.

A maximum likelihood estimator (like 3.16) is defined as the peak in the (log) likelihood of θ given some observation X. If the log likelihood is sharply peaked around $\hat{\theta}$, then it's fairly easy to pick out $\hat{\theta}$ above the noise. But if the curvature around $\hat{\theta}$ is shallow, then we're much more sensitive to effects of local noise.

This intuition is formalized in the *Fisher information*, which for a single parameter is defined

$$I(\theta) = -\mathbb{E}\left[\frac{\partial^2 \log f_X(x \mid \theta)}{\partial \theta^2} \;\middle|\; \theta\right] = -\int \frac{\partial^2 \log f_X(x \mid \theta)}{\partial \theta^2} f_X(x \mid \theta) \mathrm{d}x$$

(At least, provided that $\log f_X(x \mid \theta)$ is twice differentiable.) Intuitively, this is the expected curvature of the log likelihood function around the true parameter value, reflecting how well we can pick out the true parameter value from its close neighbors.

The CRLB is the inverse of the Fisher information. It can be shown to place a lower bound on the variance of any unbiased estimator:

$$\mathrm{CRLB}(\theta) = \frac{1}{I(\theta)} \leq \mathrm{Var}(\hat{\theta})$$

Here, we derive the Cramer-Rao lower bound (CRLB) for the MSD-based estimators 3.16 and 3.18. In both cases, we will assume that the jumps of a trajectory - localization error included - are independent, swallowing the caveats surrounding the localization error discussed earlier.

Take the estimator for the diffusion coefficient with localization error (3.18). Given the probability density 3.17, we seek the CRLB

$$I(D) = \int_{-\infty}^{+\infty} \frac{\partial^2 \log f_S(s|D)}{\partial D^2} f_S(s|D) \mathrm{d}s$$

which is

$$I(D) = \frac{Lt^2}{\left(Dt + \sigma_{\mathrm{loc}}^2\right)^2}$$

As such, the variance of the maximum likelihood estimator is

$$\mathrm{Var}(\hat{D}) = \begin{cases} \frac{\left(Dt+\sigma_{\mathrm{loc}}^2\right)^2}{Lt^2} & \text{if } \sigma_{\mathrm{loc}}^2 > 0 \\ \frac{D^2}{L} & \text{otherwise} \end{cases} \tag{3.19}$$

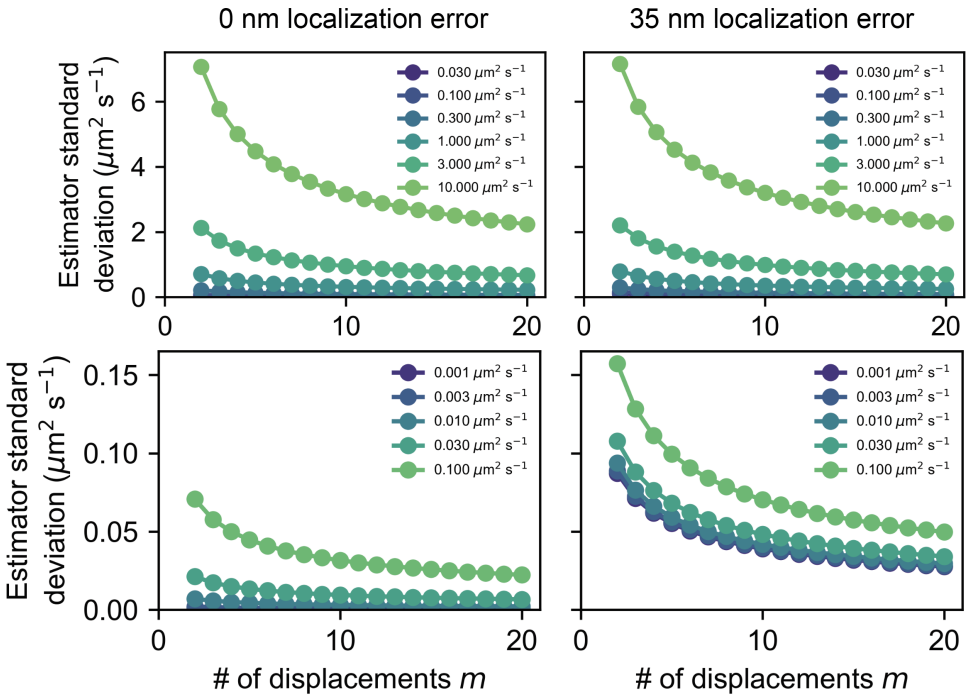

Figure 3.2: Cramer-Rao lower bound for the mean squared displacement estimator of the diffusion coefficient of a single trajectory. In this case, the frame interval was held constant at 10 ms.

where L is the number of points in this 2D trajectory.

This error is visualized in Fig. 3.2, which shows the square root of the CRLB as a function of trajectory length for several different diffusion coefficients. Notice that for the higher diffusion coefficients, the standard deviation of the estimator at 3-4 displacements (the mean trajectory length in real spaSPT datasets) is nearly equal in magnitude to the diffusion coefficient itself. Meanwhile, at lower diffusion coefficients, the accuracy of estimators becomes independent of D and is instead dominated by localization error.

If we measure in more than two dimensions - say, m dimensions - then the CRLB for the estimate of D is

$$\mathrm{Var}(\hat{D}) \geq \frac{2\left(D\Delta t + \sigma_{\mathrm{loc}}^2\right)^2}{\Delta t^2 L m}$$

Again, L is the number of jumps in the trajectory.

3.1.5 Estimators based on the jump length distribution

We've seen that, for a regular Brownian motion trajectory, the mean squared displacement is the maximum likelihood estimator when localization error is negligible. Even when localization error is present, the MSD with localization error (equation 3.18) is often useful as first-pass estimator.

Algorithm 3.1: Estimation of the diffusion coefficient and localization error for a single diffusing state in n dimensions

Parameters:

- A set of trajectories $(\mathbf{X}_1, ..., \mathbf{X}_N)$ where $(\mathbf{X}_i)_j$ is the vectorial j^{th} position of the i^{th} trajectory. Let L_i be the length of trajectory i.
- R_{max}, the maximum jump length to consider
- T, the total number of frame delays to consider
- Δr, the bin size

Algorithm:

1. For each frame gap $t = 1, ..., T$, define M_t as the total number of jumps in the dataset that occur over exactly t frames.
2. For $t = 1, ..., T$, define $H_t(r)$ as the fraction of jumps over t frames with radial displacement equal to or less than r. That is,

$$H_t(r) = \frac{1}{M_t} \sum_{i=1}^{N} \sum_{j=1}^{L_i - t} \mathbb{I}_{|(\mathbf{X}_i)_{j+t} - (\mathbf{X}_i)_j| \leq r}$$

   where $\mathbb{I}$ is the indicator function and $\left|(\mathbf{X}_i)_{j+1} - (\mathbf{X}_i)_j\right|$ is the radial distance of the j^{th} jump in trajectory i.
3. Divide the range $[0, R_{\text{max}}]$ into a set of bins $r_0 = 0$, $r_1 = \Delta r$, $r_2 = 2\Delta r$, ..., $r_K = R_{\text{max}}$.
4. Define the sum of squares function

$$S(D, \sigma^2_{\text{loc}}) = \sum_{t=1}^{T} M_t \sum_{k=0}^{K} \left(H_t(r_k) - \gamma_I \left(\frac{m}{2}, \frac{1}{4(Dt\Delta t + \sigma^2_{\text{loc}})} \right) \right)^2$$

5. Minimize $S(D, \sigma^2_{\text{loc}})$ with standard nonlinear least-squares methods (e.g. Levenberg-Marquardt, dogbox, etc.).

However, we can also attempt to fit the distribution of jumps directly, which is easier to generalize to multi-state models in the next chapter. The best way to do this is typically to fit the empirical distribution function of the observed displacements to the CDF for the radial displacements of a particle at various time

delays (equation 3.10). This approach is summarized in Algorithm 3.1. Notice in particular that estimating the localization error is only possible when considering multiple possible delays in the data.

3.1.6 Fractional Brownian motion

Fractional Brownian motion (FBM) is a Gaussian process, like RBM (Appendix B). As a result, its jump distributions extend in a straightforward manner from those discussed for RBM in section 6.1. Only the temporal scaling of the distribution changes.

In particular, if X_t is the spatial position of a one-dimensional FBM after time t, then the joint distribution of X_{t_1}, X_{t_2}, ... for any set of time indices t_1, t_2, ... is given by a multivariate normal distribution with expectation $\mathbb{E}[X_t] = 0$ and covariance defined by

$$\mathrm{Cov}(X_t, X_s) = D\left(t^{2H} + s^{2H} - |t-s|^{2H}\right)$$

Two parameters characterize the FBM. $D \geq 0$ is a scaling factor for the overall dispersion per unit time of the process and $H \in (0,1)$ is the Hurst parameter, parametrizing the dependence between the jumps. For $H < 1/2$ the jumps are anticorrelated, for $H > 1/2$ the jumps are correlated, and for $H = 1/2$ the jumps are completely independent and the process is regular Brownian motion (see the previous chapter for a complete discussion).

For all of the results in this chapter, we can use either the regular diffusion coefficient for D, which has units of $\mu\mathrm{m}^2\ \mathrm{s}^{-2H}$, or the modified diffusion coefficient discussed in Appendix B, which has units of $\mu\mathrm{m}^2\ \mathrm{s}^{-1}$ regardless of the Hurst parameter. The covariance is then

$$\mathrm{Cov}(X_t, X_s) = \overline{D}\left(t^{2H} + s^{2H} - |t-s|^{2H}\right) = \frac{D}{\Delta t^{2H-1}}\left(t^{2H} + s^{2H} - |t-s|^{2H}\right)$$

Here, Δt is the experimental frame interval.

We prefer the modified diffusion coefficient as it makes it easier to compare FBMs with different Hurst parameters for the illustrations in this section. However, as it depends on Δt, it must always be reported alongside the frame interval.

MSD-based estimators

The covariance above leads directly to the one-dimensional jump length variance

$$\mathrm{Var}(X_t) = 2\overline{D}t^{2H}$$

Since $\mathrm{Var}(X_t) = \mathbb{E}\left[X_t^2\right]$ for a mean-zero Gaussian process X_t, we have the simple mean-squared displacement

$$\mathrm{MSD}(t) = 2m\left(\overline{D}t^{2H} + \sigma_{\mathrm{loc}}^2\right)$$

where m is the number of spatial dimensions. This is a simple extension of the results for RBM in the previous sections.

A large fraction of the literature defines anomalous diffusion as any process for which the mean-squared displacement of a single particle goes as t^α with $\alpha \neq 1$. We immediately see that $H = \alpha/2$. This makes FBM a particularly attractive model for anomalous diffusion, since it inherits many of the analyticall tractable characteristics of RBM while incorporating nonlinear MSD scaling.

See Fig. B.2 in Appendix B for an illustration of the MSDs of FBMs with various Hurst parameters.

Radial jump distribution-based estimators

The covariance above leads directly to the one-dimensional jump length variance

$$\mathrm{Var}\left(X_t\right) = 2\overline{D}t^{2H}$$

As a result, we can make a simple modification to the radial jump distributions (equations 3.9 and 3.10) to obtain the jump distributions for an FBM in m dimensions:

$$\begin{aligned} f_R(r) &= \frac{2r^{m-1}\exp\left(-\frac{r^2}{4(Dt^{2H}+\sigma_{\mathrm{loc}}^2)}\right)}{\Gamma\left(\frac{m}{2}\right)\left(4(Dt^{2H}+\sigma_{\mathrm{loc}}^2)\right)^{\frac{n}{2}}} \qquad \text{(PDF)} \\ F_R(r) &= \gamma_l\left(\frac{m}{2}, \frac{r^2}{4(Dt^{2H}+\sigma_{\mathrm{loc}}^2)}\right) \qquad \text{(CDF)} \end{aligned} \tag{3.20}$$

Here, we have assumed that the process is represented by an independent FBM in each spatial dimension.

In addition, all of the other results in Section 6.1 still apply with the substitution of $Dt^{2H} + \sigma_{\mathrm{loc}}^2$ for $Dt + \sigma_{\mathrm{loc}}^2$. In addition, we can use Algorithm 3.1 for estimation of the diffusion coefficient and Hurst parameter.

Fig. 3.3 demonstrates CDFs for a FBM evaluated at several frame intervals and with three different Hurst parameters. Notice how the Hurst parameter determines the way that the dispersion of the jump distribution changes in time. But the fundamental shape of the jump distribution is itself unaltered, reflecting FBM's identity as a Gaussian process.

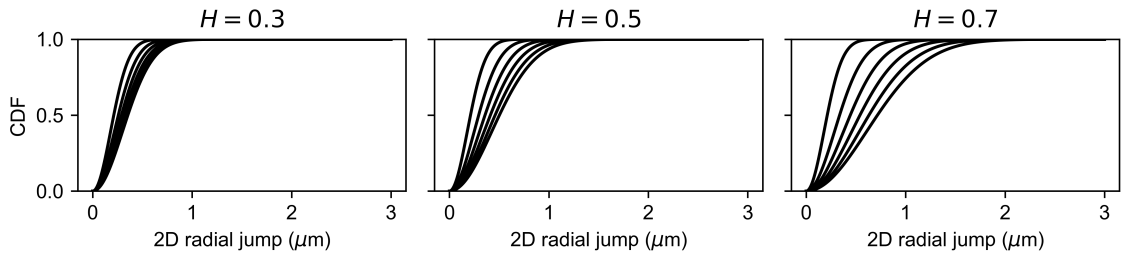

Figure 3.3: Some examples of the cumulative distribution function for the two-dimensional radial jumps of fractional Brownian motions with various Hurst parameters. Each line in the subplots represents the CDF at a subsequent frame interval: $1\Delta t$, $2\Delta t$, $3\Delta t$ and so on with $\Delta t = 0.00748$ seconds in this case. The modified diffusion coefficient was held constant at 2 μm^2 s^{-1}.

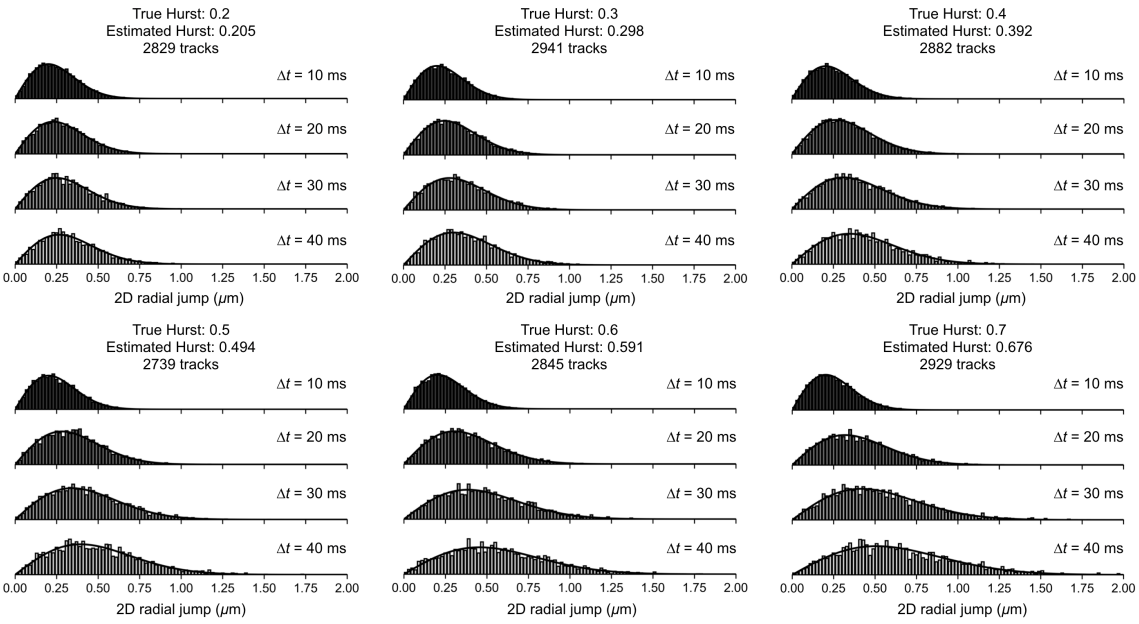

Figure 3.4: Some examples of radial jump histograms for fractional Brownian motions with various Hurst parameters, with fits. Each subplot represents the results of one simulation; the histogram are the observed jumps and the black line is the model fit. The modified diffusion coefficient was held constant at 2 μm^2 s^{-1}. Simulations were performed in a 10 μm spherical nucleus with a 700 nm focal depths, 10 ms frame intervals, 35 nm 1D localization error, and 10 Hz bleaching rate. Collisions of the particles with the walls of the nucleus were resolved by specular reflections of the jump density.

3.1.7 Levy flights

Fractional Brownian motion (FBM) is a straightforward extension of RBM. Its jumps are still distributed as Gaussians. Consequently, its estimators are simple tweaks of the estimators for RBM. But this is not the case for other anomalous diffusion models, which may depart from RBM in more fundamental ways.

In this section, we examine the case of Levy flights, which produce more interesting and varied departures from RBM. We will see that unlike RBM and RBM, Levy flights have jump distributions that are generally not separable in the geometry of most spaSPT experiments.

Jump distributions

A Levy flight in m dimensions is a random walk with independent jumps that are distributed according to a Levy stable distribution. This means that the characteristic function for the steps is

$$\phi\left(\mathbf{k}, t \mid \alpha, D\right) = \exp\left(-Dt\left|\mathbf{k}\right|^{\alpha}\right) \tag{3.21}$$

Or, if we wish to emphasize the radial symmetry, we can express it in terms of the radial distance from the origin of the Fourier domain, $k = |\mathbf{k}|$:

$$\phi\left(k, t \mid \alpha, D\right) = \exp\left(-Dtk^{\alpha}\right), \qquad k \geq 0$$

We'll see presently that only a few special cases have any sort of closed-form PDF. So we'll stick to working with characteristic functions as much as possible. One special case, $\alpha = 2$, corresponds to regular Brownian motion with diffusion coefficient D.

Equation 3.21 emerges as a solution to the fractional diffusion equation

$$\frac{\partial f(\mathbf{r}, t)}{\partial t} = D\nabla^{\alpha} f(\mathbf{r}, t) \tag{3.22}$$

where ∇^{α} is a fractional Laplacian operator, implicitly defined via its Fourier transform:

$$\begin{aligned} \mathcal{F}\left[\frac{\partial^{\alpha} f(x)}{\partial\left|x\right|^{\alpha}}\right](k) &= -\left|k\right|^{\alpha}\mathcal{F}\left[f\right](k) \qquad \text{(one dimension)} \\ \mathcal{F}\left[\nabla^{\alpha} f(\mathbf{r})\right](\mathbf{k}) &= -\left|\mathbf{k}\right|^{\alpha}\mathcal{F}\left[f\right](\mathbf{k}) \qquad (m \text{ dimensions}) \end{aligned} \tag{3.23}$$

We will tend to avoid referring to equation 3.22, but we highlight one important feature. When $\alpha = 2$, the fractional Laplacian coincides with the regular Laplacian.

For example, in 2D we have $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. However, this separation into a sum of derivatives for each spatial dimension does *not* hold more generally for other α:

$$\nabla^\alpha f(x,y) \neq \frac{\partial^\alpha f}{\partial x^\alpha} + \frac{\partial^\alpha f}{\partial y^\alpha} \qquad \text{if } \alpha \neq 2$$

If this were the case, we would have the Fourier transform

$$\mathcal{F}\left[\nabla^\alpha f\right](k_x,k_y) = -\left(|k_x|^\alpha + \left|k_y\right|^\alpha\right)\mathcal{F}\left[f\right](k_x,k_y)$$

This is physically nonsensical due to the lack of rotational symmetry in the Fourier transform when $\alpha \neq 2$. That is, we get a completely different answer if the x and y axes for $f(x,y)$ are rotated. Of course, nature does not care which direction our axes point.

As we will see, this property of the fractional Laplacian operator is at the root of most mathematical difficulties associated with Levy flights in imaging setups with finite depth of field.

Generalized central limit theorem

There are many ways to parametrize anomalous diffusion processes. But some are more useful than others. Fractional Brownian motion, with its Gaussian displacements, retains the central utility of RBM for modeling diffusion that results from many independent sources of noise with finite variance. This is a consequence of the central limit theorem. Indeed, the CLT was implicitly used to define the jump distributions of RBM in Einstein's 1905 paper. As outlined in Appendix B, section B.5, the sources of variance for FBM have a different temporal structure than RBM, which means that despite sharing the Gaussian character of RBM, FBM is capable of modeling a broad variety of processes beyond Markov processes, hence its utility to the experimentalist.

So if that's what FBMs are good for, what are Levy flights good for?

The answer comes from the generalization of the central limit theorem by Gnedenko & Kolmogorov [61]. These authors considered sums of independent random variables with potentially *infinite* variance, focusing in particular on random variables with power law tails so that $f_X(x) \propto |x|^{-\alpha-1}$.

If we gather a lot of these kind of random variables and take their mean, the result will tend to a Levy stable distribution with the characteristic function $\exp\left(-c\,|k|^\alpha\right)$, where c is a dispersion parameter and α is the *stability parameter*. α is fundamentally related to the character of the underlying random variables. The fatter the tails of the PDF for the constituent random variables, the lower α becomes and

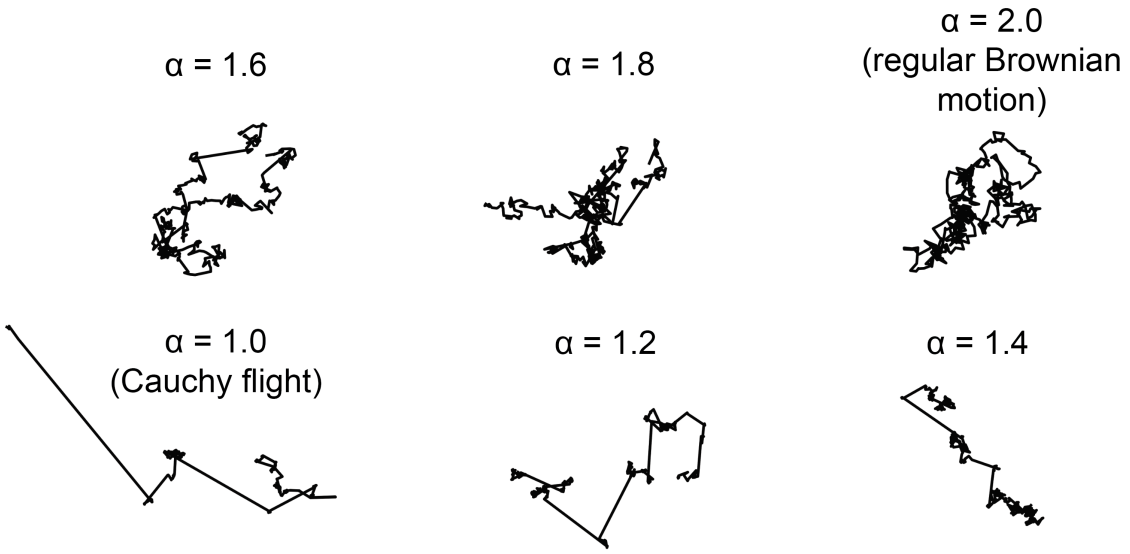

Figure 3.5: Some Levy flights with different stability (α) parameters. Each Levy flight has the same dispersion parameter *D* and have been projected from their native 3D space onto a 2D plane.

the "wilder" the resulting motion is.

If we assume that the dispersion scales linearly in time, we recover the characteristic function for Levy flights (equation 3.21).

The natural emergence of Levy flights as a consequence of the generalized central limit theorem is perhaps responsible for their observation in a wide range of fields, from the fluctuations of stock-market prices [62] [63] to the paths of foraging animals [64]. Having random variables with infinite variance is not as exotic as it may seem; in fact, heavy-tailed probability distributions with this property seem to be the norm rather than the exception in biological research.

The effect of these types of noise is that Levy flights transition between exploration of local spatial neighborhoods and longer jumps between neighborhoods. The balance between local exploration and longer jumps is set by the stability parameter. As $\alpha \to 2$, local exploration dominates while when $\alpha \to 0$, the longer jumps become dominant.

Fig. 3.5 demonstrates this effect. Note that as α becomes lower, the motion becomes more erratic, prone to overexploring local neighborhoods and jumping between distant neighborhoods.

Special cases of Levy flights

Levy flights have a serious problem from a practical perspective: closed-form PDFs corresponding to the characteristic function 3.21 only exist for three specific values of α. We've already investigated one of them - regular Brownian motion - in detail. Clearly, these special cases are not going to carry us very far, but examining them can give some insight into the practical significance of the stability parameter.

When $\alpha = 2$, we have a Gaussian random walk, equivalent to a regular Brownian motion observed at a discrete set of timepoints. The jump length distributions for this type of motion were discussed extensively in the first section of this chapter.

When $\alpha = 1$, we have Cauchy motion. The characteristic function is $\exp\left(-Dt\left|\mathbf{k}\right|^2\right)$. Comparing with equation C.11 in Appendix C, we can draw out the real-space density

$$f_{\mathbf{R}}(\mathbf{r}) = \frac{\Gamma\left(\frac{m+1}{2}\right)}{\pi^{\frac{m+1}{2}}(Dt)^m\left(1+\left(\frac{|\mathbf{r}|}{Dt}\right)^2\right)^{\frac{m+1}{2}}} \tag{3.24}$$

Substituting $m = 1$, we recover the familiar univariate Cauchy density with dispersion parameter Dt.

Both the expectation and variance of the Cauchy jump density are undefined. To see this, let's examine the one-dimensional case more closely. Applying the moment property C.2 from Appendix C,

$$i\mathbb{E}\left[X\right] = \frac{\partial\phi_X(k)}{\partial k}\Big|_{k=0} = \frac{\partial}{\partial k}\exp\left(-Dt\left|k\right|\right)\Big|_{k=0}$$

But

$$\frac{\partial}{\partial k}\exp\left(-Dt\left|k\right|\right) = \begin{cases} e^{-Dt|k|} & \text{if } k > 0 \\ -e^{-Dt|k|} & \text{if } k < 0 \\ \text{undefined} & \text{if } k = 0 \end{cases}$$

As a result, the expectation of the Cauchy density is undefined. Indeed, this property extends to all Levy stable variables with $\alpha \leq 1$. To see this, consider the first derivative of the characteristic function 3.21:

$$\frac{\partial\phi_X(k)}{\partial k} = \alpha\text{sgn}(k)\left|k\right|^{\alpha-1}e^{-|k|^\alpha}$$

Since this only has a unique limit at $k = 0$ for $\alpha > 1$, it is only at these values that the jumps of a Levy flight have finite expectation.

Likewise, differentiating the characteristic function twice gives us

$$\frac{\partial^2 \phi_X(k)}{\partial k^2} = \alpha D t \left(-(\alpha - 1) |k|^{\alpha - 2} + \alpha D t |k|^{2(\alpha - 1)} \right) e^{-Dt|k|^\alpha}$$

This is only continuous at $k = 0$ when $\alpha = 2$. As a result of equation C.2, this means that the variance of the jump density is not defined for any $\alpha < 2$.

The last special case is $\alpha = 1/2$, which corresponds to a random walk where the steps have a Levy distribution. (The name is unfortunate. This should not be confused with the more general class of Levy stable distributions.) While this has a closed-form PDF for the univariate case, in general no solution exists for the multivariate case and so we refrain from discussing it, in part because the ambiguous naming produces serious headaches.

Radial jump distributions of Levy flights

The previous section poses a conundrum. Levy flights are a useful model for spaSPT analysis because they are in a sense the simplest possible distributions that incorporate sources of noise wilder than the finite-variance noise that leads to Brownian motion. In short, they provide valuable models when the sources of noise have a different character than those that Einstein assumed in his 1905 paper.

But the problem is that we can only get closed-form solutions for the PDF in two special cases: $\alpha = 2$ and $\alpha = 1$. Apart from these two cases, the situation is much more difficult. In order to extract parameters for these types of motion from biological data, we need efficient numerical methods to evaluate the jump PDFs. Since radial jump PDFs are most useful from the perspective of modeling, we focus on obtaining on these, rather than Cartesian densities.

First we'll look at the naive approach, where we simply take the inverse Fourier transform of the characteristic function 3.21. Then we'll examine a more elegant, faster, and more accurate approach.

In what follows, we'll continue to use $f_{\mathbf{R}}(\mathbf{r})$ to denote the jump PDF represented in Cartesian coordinates and $f_{\mathbf{R}}(r)$ to denote the same PDF represented in terms of the radial distance from the origin $r = \sqrt{x^2 + y^2 + z^2}$.

Naive approach

One approach is simply to take the inverse transform of the Levy flight character-

istic function 3.21, then marginalize on all of the angular components:

$$f_{\mathbf{R}}(\mathbf{r}) = \mathcal{F}_m^{-1}\left[\phi_{\mathbf{R}}(\mathbf{k})\right]$$
$$f_R(r) = \int_0^{2\pi} \mathrm{d}\psi \int_0^{\infty} \mathrm{d}\theta \int_0^{\infty} \mathrm{d}r\, r^2 \sin\theta\, f_{\mathbf{R}}(r) \qquad (m = 3)$$

Here, $\mathcal{F}_m^{-1}$ is the inverse Fourier transform in n dimensions. Using the relation C.10 from Appendix C, and assuming that the process natively happens in 3D, we can write the radial density as

$$f_R(r) = \frac{1}{(2\pi)^{\frac{3}{2}} r^{\frac{1}{2}}} \mathcal{H}_{\frac{1}{2}}\left[k^{\frac{1}{2}} e^{-Dt|k|^{\alpha}}\right](r)$$

where $\mathcal{H}_{\frac{1}{2}}$ is a Hankel transform of order $1/2$. So this approach boils down to taking a Hankel transform. This is feasible, but computationally demanding to do at high accuracy and throughput. It also tends to Gibbs phenomena near the origin when the parameters for numerical integration are chosen carelessly.

Radon transform approach

An alternative route to evaluating the jump density for a Levy flight comes from the operator cycle C.17, which can be written

$$\mathcal{F}_1\left[\mathcal{R}\left[f_{\mathbf{R}}\right]\right](k) = \mathcal{F}_n\left[f_{\mathbf{R}}\right](k)$$

$\mathcal{R}$ is the Radon transform, defined in 3D as

$$\hat{f}(p, \mathbf{v}) = \mathcal{R}\left[f_{\mathbf{R}}\right] = \iiint_{-\infty}^{\infty} f_{\mathbf{R}}(\mathbf{r})\, \delta(p - \mathbf{v}\cdot\mathbf{r})\mathrm{d}\mathbf{r}$$

As discussed in Appendix C, this operator projects the density $f_{\mathbf{R}}$ onto a line in the direction $\mathbf{v}$. When the density is radially symmetric, we can choose whatever $\mathbf{v}$ we like.

In our case, we don't know the real domain function $f_{\mathbf{R}}$ and start instead knowing $\phi_{\mathbf{R}}(k) = \mathcal{F}_n\left[f_{\mathbf{R}}\right] = e^{-Dtk^{\alpha}}$. So our problem is essentially to invert the Radon transform:

$$f_{\mathbf{R}}(r) = \mathcal{R}^{-1}\left[\mathcal{F}_1^{-1}\left[\phi_{\mathbf{R}}\right]\right]$$

First, we note that for a radially symmetric 3D function, we can choose $\mathbf{v} = (1, 0, 0)$

to make things simpler. Then we can express the Radon transform as

$$\begin{aligned}\hat{f}_{\mathbf{R}}(p) = \mathcal{R}\left[f_{\mathbf{R}}\right] &= \int_{-\infty}^{\infty}\!\!\iiint f_{\mathbf{R}}\left(\sqrt{x^2+y^2+z^2}\right)\delta(p-x)\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z \\ &= \int_{-\infty}^{\infty}\!\!\iint f_{\mathbf{R}}\left(\sqrt{p^2+y^2+z^2}\right)\,\mathrm{d}y\,\mathrm{d}z\end{aligned}$$

Let $r_2^2 = y^2 + z^2$. Then this integral can be expressed in polar coordinates as

$$\hat{f}_{\mathbf{R}}(p) = 2\pi \int_0^{\infty} f_{\mathbf{R}}\left(\sqrt{r_2^2+p^2}\right) r_2\,\mathrm{d}r_2$$

Noting that p is simply a constant for the right side, let $r^2 = r_2^2 + p^2$ so that $r_2 \mathrm{d}r_2 = r\mathrm{d}r$. Then, changing variables, the integral becomes

$$\hat{f}_{\mathbf{R}}(p) = 2\pi \int_p^{\infty} f_{\mathbf{R}}(r) r\,\mathrm{d}r, \qquad p > 0$$

Differentiation of both sides yields

$$\frac{\partial \hat{f}_{\mathbf{R}}}{\partial p} = -2\pi p\, f_{\mathbf{R}}(p)$$

Recognizing that p is a dummy variable, we can then write the PDF as

$$f_{\mathbf{R}} = -\frac{1}{2\pi r}\frac{\partial \hat{f}_{\mathbf{R}}}{\partial r}$$

An interesting sidenote, remarked by Barrett [65], is that this result was derived several times independently - by Vest and Steel in the context of optics in 1978 [66], by Du Mond in the context of Compton scattering in 1929 [67], and by Stewart (1957) and Mijnarends (1967) in the context of positron annihilation.

Now, since the operator cycle gives us $\hat{f}_{\mathbf{R}}(r) = \mathcal{F}_1^{-1}\left[\phi_{\mathbf{R}_1}(k)\right]$, we have

$$\begin{aligned}f_{\mathbf{R}}(r) &= -\frac{1}{2\pi r}\frac{\partial}{\partial r}\mathcal{F}_1^{-1}\left[\phi_{\mathbf{R}_1}(k)\right] \\ &= -\frac{1}{2\pi r}\mathcal{F}_1^{-1}\left[ik\phi_{\mathbf{R}_1}(k)\right]\end{aligned}$$

Here, we've used the fact that differentiation in the real domain corresponds to multiplication by *ik* in the Fourier domain. Now the PDF for 3D radial displacements $f_R(r)$ is given by

$$f_R(r) \propto r^2 f_{\mathbf{R}}(r) \propto -ir\mathcal{R}_1^{-1}\left[ik\phi_{\mathbf{R}}(k)\right]$$

The proportionality holds to within a normalization constant. Multiplication by *ir* in the real domain corresponds to differentiation in the Fourier domain, from which we obtain

$$f_R(r) \sim \mathcal{F}_1^{-1}\left[\frac{\partial}{\partial k}\left(k\phi_{\mathbf{R}}(k)\right)\right] \tag{3.25}$$

Substituting the definition for the Levy flight characteristic function 3.21, this evaluates explicitly to

$$f_R(r) \propto \mathcal{F}_1^{-1}\left[\left(1-\alpha D t\,|k|^{\alpha}\right)e^{-Dt|k|^{\alpha}}\right] \tag{3.26}$$

Equation 3.26 represents the fastest and most accurate method we are aware of for calculating the probability of the 3D radial jumps of a Levy flight. In practice, we approximate this PDF by sampling it at a series of finely spaced points, then normalize over this finite support.

Fig. 3.6 shows the radial jump distributions of Levy flights with various stability parameters in different spatial dimensions. As α decreases, the overall mobility

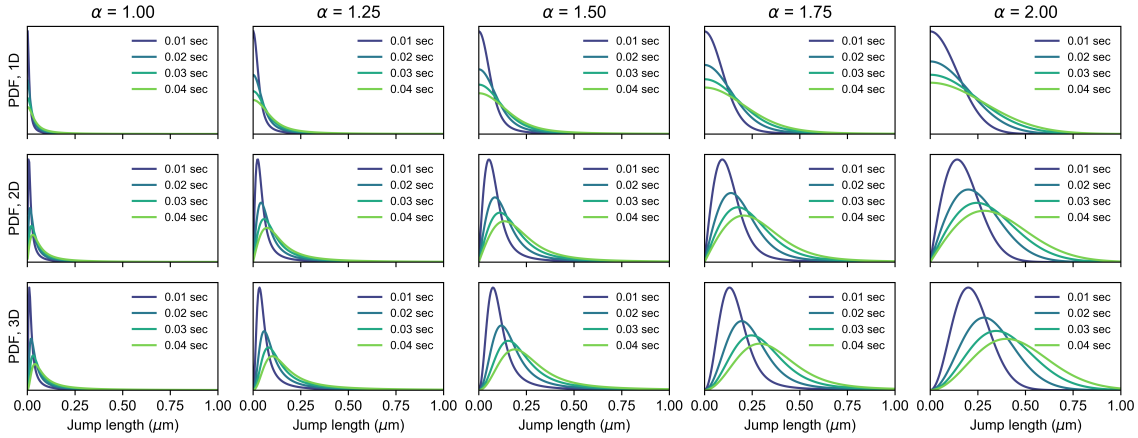

Figure 3.6: Some radial jump distributions for Levy flights with various stability parameters. All Levy flights have the same dispersion parameter $D = 1\ \mu\text{m}^{\alpha}\text{s}^{-1}$ and the jumps are measured with zero localization error in 3D at 10 ms frame intervals. The upper row is 1D jumps, the middle row is 2D jumps, and the lower row is 3D jumps. As discussed in the text, marginalization of a higher-dimensional Levy flight jump distribution only becomes a lower-dimensional distribution when the focal depth is infinite.

is dominated by a few long-distance jumps between neighborhoods, followed by overexploration of individual neighborhoods.

It is tempting to take this another step further, calculating the CDF by introducing a $(ik)^{-1}$ in the Fourier transform, then normalizing in the real domain. However, due to ensuing discontinuities at the origin, this approach is numerically unfeasible. Instead, we can approximate the CDF by numerically accumulating the PDF at finely spaced points in the real domain.

Levy flights with localization error

One of the advantages of the numerical approach introduced in the previous section is that we can easily incorporate localization error.

Suppose that $\mathbf{R}$ is the jump of a Levy flight over some finite time interval t. We've seen that this jump has the characteristic function

$$\phi_{\mathbf{R}}(\mathbf{k}) = \exp\left(-Dt\,|\mathbf{k}|^{\alpha}\right)$$

Now imagine each endpoint of the jump has some normally distributed localization error with mean zero and one-dimensional variance σ_{loc}^2. Since the error at both endpoints is independent, the one-dimensional variance contributed to the jump is $2\sigma_{\text{loc}}^2$. We can incorporate this by modeling the jump as $\overline{\mathbf{R}} = \mathbf{R} + \mathbf{X}$, where

$$\mathbf{X} \sim \mathcal{N}\left(\mathbf{0}, 2\sigma_{\text{loc}}^2 I\right)$$

Here, I is the identity matrix.

Using the convolution property C.6, $\overline{\mathbf{R}}$ has the characteristic function

$$\begin{aligned}
\phi_{\overline{\mathbf{R}}}(\mathbf{k}) &= \phi_{\mathbf{R}}(\mathbf{k})\phi_{\mathbf{X}}(\mathbf{k}) \\
&= \exp\left(-Dt\,|\mathbf{k}|^{\alpha} - \sigma_{\text{loc}}^2\,|\mathbf{k}|^2\right) \\
&= \exp\left(-Dtk^{\alpha} - \sigma_{\text{loc}}^2 k^2\right), \qquad k \geq 0
\end{aligned}$$

In the last equation, we have emphasized that this characteristic function is only a function of the radial distance from the origin $k = |\mathbf{k}|$. As a result, we can directly apply equation 3.25 to numerically compute the 3D radial displacement PDF.

2D radial jumps

Suppose we have a Levy flight in 3D. The characteristic function for the jumps is given by equation 3.21. Suppose we project these jumps out of their native 3D

onto a 2D surface - for instance, the surface of a camera.

For clarity, let $\mathbf{R}$ be the Cartesian coordinates of the jump, let R_3 be the radial distance from the origin in 3D, and let R_2 be the radial distance from the origin in the 2D projection plane.

Then, applying the Fourier slice theorem C.8,

$$\begin{aligned} f_{R_2}(r) &= \mathcal{F}_2^{-1}\left[\phi_{\mathbf{R}}(k)\right](r) \\ &= \frac{1}{2\pi}\mathcal{H}_0\left[\phi_R(k)\right](r) \end{aligned}$$

where $\mathcal{H}_0$ is the zeroth order Hankel transform (see equation C.10).

In practice, it is actually easier to start from a 3D radial density as computed by 3.25, then apply the Abel transform to shift down by a dimension:

$$\begin{aligned} f_{R_2}(r) &= \mathcal{A}\left[\mathcal{F}_1^{-1}\left[\frac{\partial}{\partial k}\left(k\phi_{\mathbf{R}}(k)\right)\right]\right] \\ &= \mathcal{A}\left[ir\mathcal{F}_1^{-1}\left[k\phi_{\mathbf{R}}(k)\right]\right] \end{aligned}$$

There's a problem here for real experiments though. Imagine that we're specifically projecting a Levy flight onto the lateral XY plane of a camera. In essence, the Fourier slice theorem integrates over the jump distribution in the axial Z direction. Implicit in the use of the slice theorem is that the integration bounds are at $\pm\infty$.

Suppose instead that we can only observe particles in a thin slice in the axial direction with thickness Δz. This is the case, for instance, in any imaging situation with a finite depth of field. The depth of field is especially shallow when using the high-NA objectives required for spaSPT experiments - - about 700 nm in our experiments. Imagine that our particle starts at $z = 0$, so that the boundaries of observation are at $\pm\Delta z/2$.

We can attempt to derive the Fourier slice theorem in this geometry, following

the general pattern of Section C.4 in Appendix C. Marginalizing over z,

$$\begin{aligned} f_{\mathbf{R}_2}(x,y) &= \int_{-\Delta z/2}^{+\Delta z/2} \mathrm{d}z\, f_{\mathbf{R}_3}(x,y,z) \\ &= \int_{-\Delta z/2}^{+\Delta z/2} \mathrm{d}z\, \frac{1}{(2\pi)^3} \iiint_{-\infty}^{\infty} \mathrm{d}\mathbf{k}\, \phi_{\mathbf{R}_3}(k_x,k_y,k_z) e^{-i(k_x x + k_y y + k_z z)} \\ &= \frac{1}{(2\pi)^3} \iiint_{-\infty}^{\infty} \mathrm{d}\mathbf{k}\, \phi_{\mathbf{R}_3}(k_x,k_y,k_z) e^{-i(k_x x + k_y y)} \int_{-\Delta z/2}^{+\Delta z/2} \mathrm{d}z\, e^{-ik_z z} \end{aligned}$$

This gives

$$f_{\mathbf{R}_2}(x,y) = \frac{\Delta z}{(2\pi)^2} \iiint_{-\infty}^{\infty} \mathrm{d}\mathbf{k}\, \phi_{\mathbf{R}_3}(\mathbf{k}) \,\mathrm{sinc}\left(\frac{k_z \Delta z}{2\pi}\right) e^{-i(k_x x + k_y y)} \tag{3.27}$$

where

$$\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

Only when $\Delta z \to \infty$ does this resolve into the slice theorem.

Suppose, further, that $\phi_{\mathbf{R}_3}(\mathbf{k})$ is inseparable, meaning we cannot factor it into components along each spatial dimension:

$$\phi_{\mathbf{R}_3}(\mathbf{k}) \neq \phi_{X,Y}(k_x,k_y)\phi_Z(k_z)$$

Placing this into equation 3.27, a rather unexpected result emerges: *the 2D radial jump distribution in this projection will not be the same as the 2D radial jump distribution in free space.*

This result is profound. It means that, unless we are able to observe jumps across the entirety of the z axis, the distribution of XY displacements of jumps will be dependent on our imaging geometry. Moreover, the jump distribution in the XY plane will *depend on the exact position that the particle started in Z.*

This is remarkable because we started with a true Markov process: a Levy flight in 3D, with a fully radially symmetric jump distribution. The result has a jump distribution that depends on where the particle starts in z. In essence, the imaging geometry transforms a Markov diffusion process into a non-Markov diffusion process.

In order to accurately compute the jump distribution observed in the plane of the camera, we need to account for this finite depth of field. Our approach is the following:

1. Calculate the 3D radial jump distribution using the fast method defined by 3.25 on the characteristic function $\exp\left(-Dtk^{\alpha} - \sigma_{\text{loc}}^2 k^2\right)$.
2. Project out of a 3D slice with finite thickness Δz onto a 2D plane using the *finite depth Abel transform*.

We define the *finite depth Abel transform* in the following way. Suppose we have a particle that begins at the position $(0, 0, Z_0)$ and then makes a jump $\mathbf{R}_3 = (X, Y, Z)$. Because the axial dimension is hidden to us in the SPT experiment, we only measure $\mathbf{R}_2 = (X, Y)^T$. Further, we only actually observe this jump if it ends up in the focal volume. That is, we only observe the jump if $Z_0 \in \left[-\frac{\Delta z}{2}, \frac{\Delta z}{2}\right]$ and $Z_0 + Z \in \left[-\frac{\Delta z}{2}, \frac{\Delta z}{2}\right]$.

If we let $R_2 = \sqrt{X^2 + Y^2}$, then we can represent the distribution of R_2 conditional on some starting axial position Z_0 as

$$f_{R_2|Z_0}(r_2|z_0) = \int_{-\Delta z/2 - z_0}^{\Delta z/2 - z_0} f_{R_3}\left(\sqrt{r_2^2 + z^2}\right) \mathrm{d}z$$

But we don't know the starting position z_0. Assuming complete ignorance, we'll give Z_0 a uniform distribution from $-\Delta z/2$ to $\Delta z/2$. Then, marginalizing out Z_0, we have

$$\mathcal{A}_{\Delta z}\left[f_{R_3}\right] = f_{R_2}(r_2) = \frac{1}{\Delta z} \int_{-\Delta z/2}^{\Delta z/2} \mathrm{d}z_0 \int_{-\Delta z/2 - z_0}^{\Delta z/2 - z_0} \mathrm{d}z\, f_{R_3}\left(\sqrt{r_2^2 + z^2}\right) \tag{3.28}$$

This will be our definition of the finite-depth Abel transform.

To compare with the Abel transform C.13, we can also rearrange the finite-depth Abel transform as

$$f_{R_2}(r_2) = \frac{1}{\Delta z} \int_{-\Delta z/2}^{\Delta z/2} \mathrm{d}z_0 \int_{\sqrt{r_2^2 + (\Delta z/2 + z_0)^2}}^{\sqrt{r_2^2 + (\Delta z/2 - z_0)^2}} \frac{f_{R_3}(r_3) r_3}{\sqrt{r_3^2 - r_2^2}} \mathrm{d}r_3$$

Here, the subscripts in r_3 and r_2 emphasize the role they play in the geometry. While somewhat intimidating, the idea of this transform is simple. It represents the

actual process by which the 3D path of our particle in the focal depth is projected onto the 2D surface of the camera. When the jump distribution is separable in the spatial dimensions so that

$$f_{R_3}\left(\sqrt{r_2+z^2}\right) = f_{R_2}\left(r_2\right) f_Z\left(z\right)$$

then the finite-depth Abel transform just becomes

$$\mathcal{A}_{\Delta z}\left[f_{R_3}\right] = (\text{constant term}) \cdot f_{R_2}(r_2)$$

This is the case, for example, for regular or fractional Brownian motion. As we have seen, it is not the case for Levy flights due to the inseparability of their 3D density.

While the finite-depth Abel transform is not usually possible to solve analytically, it does lend itself to a fast numerical algorithm to transform 3D histograms into 2D histograms (Algorithm 3.2). This is sufficient for most practical purposes, including model fitting.

Algorithm 3.2: Numerical finite-depth Abel transform

Purpose:

Projects a distribution of 3D radial displacements into a distribution of 2D radial displacements, given a finite depth of field Δz and a random starting position in z.

Parameters:

- Δz, the focal depth
- $R_{\max}$, the maximum jump length to consider
- Δr, the bin size

Precomputations:

Allocate a histogram $\mathbf{H}$. The element $H_{j,i}$ corresponds to jumps that have a 3D radial displacement R_3 that falls into the range $[i\Delta r, (i+1)\Delta r)$ and a 2D radial displacement R_2 that falls into the range $[j\Delta r, (j+1)\Delta r)$.

For some suitably high number of iterations N, sample in the following way:

1. For each bin $i = 0, 1, 2, \ldots$ such that $0 \leq i\Delta r < R_{\max}$:
   (a) Pick an point (X, Y, Z) with uniform probability from the surface of the unit sphere.
   (b) Generate a random number $Z_0 \sim \text{Uniform}\left(-\frac{\Delta z}{2}, \frac{\Delta z}{2}\right)$.
   (c) Generate a random number $u \sim \text{Uniform}(0, 1)$, then determine the radius
   $$R_3 = \left(r_0^3 + u\left((r_0 + \Delta r)^3 - r_0^3\right)\right)^{1/3}$$
   (d) If $R_3 Z + Z_0 \in [-\Delta z/2, +\Delta z/2]$, then determine $R_2 = \sqrt{X^2 + Y^2}$ and place it in the corresponding bin of $\mathbf{H}$. Find the j such that $R_2 \in [j\Delta r, (j+1)\Delta r]$ and increment the corresponding $H_{j,i}$ by $1/N$.

The array $\mathbf{H}$, which is a matrix operator corresponding to the finite-depth Abel transform, can be saved for later use.

Algorithm: Represent a 3D radial jump distribution as a vector $\mathbf{v}$, where the element v_i is the fraction of jumps that fall into the interval $[i\Delta r, (i+1)\Delta r)$. Then the distribution of 2D jumps is $\mathbf{Hv}$. Additionally, $1 - \sum_j (\mathbf{Hv})_j$ is is the fraction of jumps that do not land within the focal volume.

Since it takes some time to precompute the matrix operator $\mathbf{H}$, Algorithm 3.2 limits us to working with jump histograms with a maximum displacement R_{max} and bin size Δr that are known in advance. This is usually not a problem, provided R_{max} is chosen to be much larger than any experimentally observed displacement. In our case, we usually choose $R_{\text{max}} = 20$ μm.

Fig. 3.7 shows some examples of this approach applied to measure the stability parameter α for various Levy flights. In this case, jumps from 10 ms to 40 ms were used for fitting, which produces reasonably accurate estimates of α.

3.1.8 Summary

Here, we briefly review some of the most important results from this part of the chapter:

1. The mean-squared displacement (MSD) provides a fast and easy approximation to the maximum likelihood estimator for RBM with localization error.

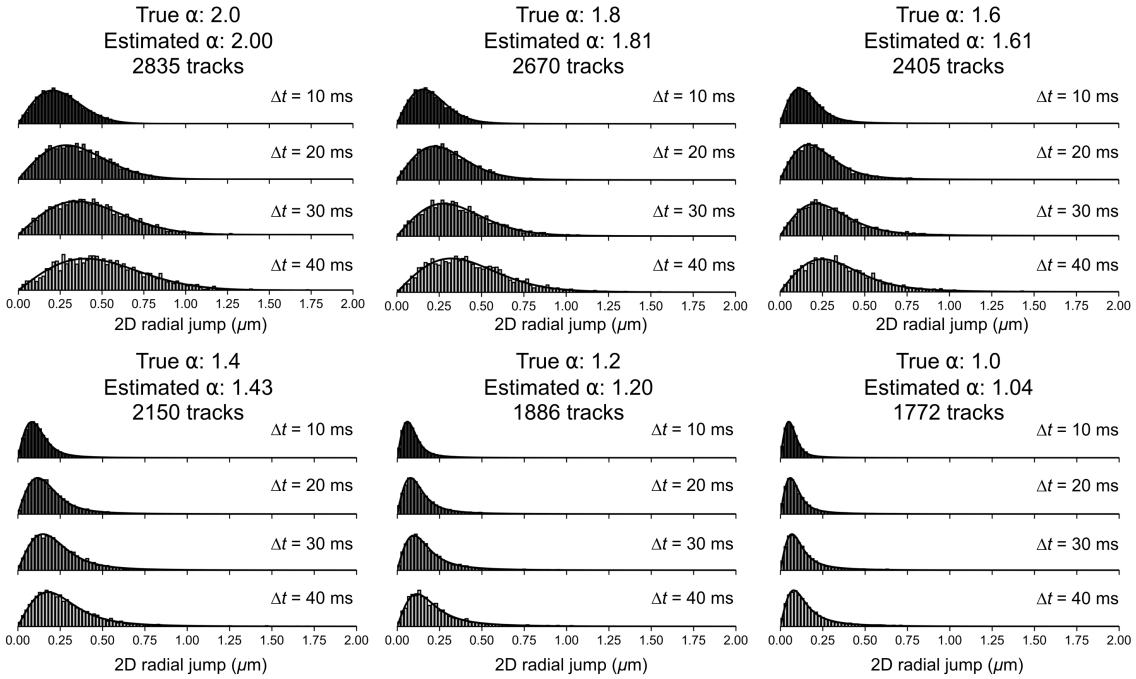

Figure 3.7: Using the radial jump histogram to extract the stability parameter from simulated Levy flights. In all cases, the dispersion parameter was kept constant at $D = 2.0$ μm^{α} s^{-1}. Histograms are observed jumps, black lines are model fits. Simulations were performed in a 10 μm spherical nucleus with a 700 nm focal depth, 10 ms frame intervals, 35 nm 1D localization error, and a 10 Hz bleaching rate. The probability density for individual jumps interacts with the nuclear boundaries via specular reflections.

The approximation becomes exact when localization error is zero.

2. Due to localization error, even immobile objects appear to diffuse in spaSPT data.
    - When considering single jumps, the apparent diffusion coefficient due to localization error is $D_{\text{err}} = \sigma_{\text{loc}}^2/\Delta t$, where σ_{loc}^2 is the 1D variance associated with the localization method and Δt is the frame interval.
    - For RBM and FBM, diffusion cannot be distinguished from localization error without considering jumps over multiple frame intervals.
3. Parameters for non-RBM diffusion models, such as FBM and Levy flights, can be extracted to fitting the empirical distribution function ("CDF") for the radial jumps.
    - In the case of FBM, it is also possible to extract model parameters via the MSD relation $\text{MSD}(t) \propto Dt^{2H}$, although the jump length histogram is preferable because it generalizes more readily to mixture models.
4. For separable diffusion models like RBM and FBM, we can treat diffusion in the plane of the camera as intrinsically two-dimensional. This is *not* the case for inseparable models like Levy flights, which require that we consider the finite focal depth of the spaSPT setup.
5. Fast algorithms to calculate the PDF and CDF for Levy flights exist based on the finite-depth Abel transform.

The models considered in this chapter provide the building blocks for more complex diffusion models. In the next chapter, we construct some of these by combining individual diffusive states into *mixtures*.

3.2 Identifying the type of motion

The previous section provided methods to parametrize regular Brownian motion (RBM) and two kinds of departure from it - fractional Brownian motion (FBM) and Levy flights. These estimators work well when dealing with single diffusing states where we have a reasonably well defined idea about the mode of diffusion. However, assumptions in model selection - for instance, that diffusion is normal, that localization error can be neglected, or that each molecule can be treated as a sample from the same distribution - can have strong consequences for inference. In later chapters, we examine diffusion model inference techniques that attempt to remove some of these assumptions.

This chapter examines a different, and earlier, aspect of spaSPT analysis. Before spaSPT data is interpreted with diffusion models, it should first be subjected to simple nonparametric methods that interrogate qualitative aspects of the mode of motion. These methods can guide subsequent choices for the type of model to use with a given spaSPT dataset.

In the course of these experiments, two of the most important questions before proceeding to any model-dependent analysis are:

1. How many distinct types of motion are present in the dataset?
2. What is the type of motion? (Is there evidence of subdiffusion or superdiffusion?)

Of course, it is also possible to interpret spaSPT data with a model first, then discriminate between models later. A common way to discriminate between models is the Akaike information criterion or the Bayesian information criterion, which are based on essentially arbitrary penalties for the number of model parameters. (Indeed, there exist many situations for which the AIC and BIC give opposite answers.) A more principled approach from Bayesian statistics is the maximum evidence method [48], which is the criterion used for model selection in the variational Bayesian framework vbSPT [50] and in the variational Bayes methods described later in this thesis (chapter 5). However, when applied to discriminate between normal and anomalous diffusion models, the maximum evidence method depends heavily on the specific parametrization and can sometimes result in sharp, sudden transitions between "normal" and "anomalous" regimes. Nature is under no constraint to be so categorical. As such, even when the maximum evidence method is used, simple visual nonparametric ways to find anomalous diffusion are still valuable aids, and they become essential at the stage when parametrization of the process is still incomplete.

In SPT experiments with exactly one diffusive state, linearity of the mean squared displacement (MSD) with respect to time is an excellent example of a nonparametric way to identify anomalous diffusion. As we saw in the introduction, however, the MSD is nearly useless in the presence of multiple diffusing states when the microscope's depth of field is finite. Since this is the case in essentially every live cell spaSPT experiment performed to date, we must seek other methods.

First we discuss angular distributions, which may be seen as a direct alternative to MSDs for identifying subdiffusion and superdiffusion. In the course of this discussion, however, we demonstrate that any test based on the detection of Markov (memoryless) dynamics is a flawed predictor of anomalous diffusion due to the ubiquity of localization error. Indeed, localization error delivers a death blow to

the Markov property of all Gaussian processes, encompassing many diffusion processes beyond the ones considered in this chapter. Finally, we discuss a method that paints a more general portrait of anomalous diffusion.

3.2.1 Angular distributions

An attractive alternative to the MSD is the angular distribution, which is schematized in Fig. 3.8. In particular the angular distribution can be related directly to the Hurst parameter of a fractional Brownian motion (FBM), which opens the door to the powerful techniques associated with FBM and more generally Gaussian processes.

Motivated by this application, we'll examine the angular distribution through the lens of FBM. First, we derive the angular distribution for an FBM with any Hurst parameter. Next, we construct a Gaussian process closely related to an FBM that incorporates localization error, an unavoidable aspect of real spaSPT data with strong consequences for the memory effects of an FBM. Finally, we discuss the MSD and the angular distributions as parametrizations of a Gaussian process covariance function, which can be seen as a more general alternative to either of these methods.

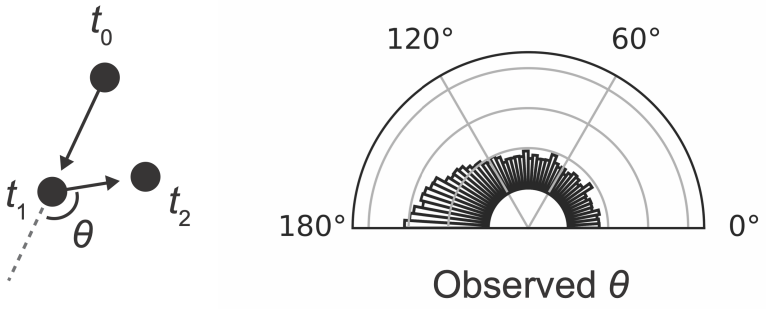

Figure 3.8: Schematic of the angle θ between subsequent displacements in a trajectory. The histogram on the right is the distribution of experimentally observed angles for H2B-HaloTag in U2OS nuclei labeled with the dye PA-JFX549 and imaged at 7.48 ms frame intervals, considering only displacements equal to or greater than 160 nm in length.

3.2.2 Angular distribution for processes with long-range memory

We consider the angular distribution of a 2D FBM with Hurst parameter H and diffusion coefficient D. A 2D FBM is constructed simply by the combination of two orthogonal 1D FBMs: diffusion along the x and y axes is assumed to be given by two independent stochastic processes X_t and Y_t such that $t \in \mathbb{R}$ and

$$\begin{aligned}
\mathbb{E}[X_t] &= 0 \\
\mathbb{E}[Y_t] &= 0 \\
\mathrm{Cov}(X_t, X_s) &= D\left(t^{2H} + s^{2H} - |t-s|^{2H}\right) \\
\mathrm{Cov}(Y_t, Y_s) &= D\left(t^{2H} + s^{2H} - |t-s|^{2H}\right)
\end{aligned}$$

Along with the mean and covariance functions, we assume that the joint distribution of the stochastic process at any finite set of points $t_1, ..., t_n$ is given by the multivariate normal density. This assumption qualifies the 2D FBM as a Gaussian process. (See Appendix B for a full discussion).

Consider three points from this process: (X_0, Y_0), $(X_{\Delta t}, Y_{\Delta t})$, and $(X_{2\Delta t}, Y_{2\Delta t})$. Given the length of the jump between the first two points, we seek the distribution of angles formed by these three points in the XY plane.

From the definition above we have $X_0 = 0$ and $Y_0 = 0$. Further, let $X_{\Delta t} = x_1$ and $Y_{\Delta t} = 0$. We can do the latter since the distribution is isotropic, so we can always rotate the three points so that the first jump aligns with the x axis without affecting the angle.

The next step is to find the joint distribution of $(X_{2\Delta t}, Y_{2\Delta t})$. Here, we exploit the useful conditioning property of Gaussian processes (equation B.3, described in Appendix B) to find that

$$\begin{aligned}
X_{2\Delta t} \mid (X_{\Delta t} = x_1) &\sim \mathcal{N}\left(2^{2H-1}x_1, D\Delta t^{2H}(2^{2H+1} - 2^{4H-1})\right) \\
Y_{2\Delta t} \mid (Y_{\Delta t} = 0) &\sim \mathcal{N}\left(0, D\Delta t^{2H}(2^{2H+1} - 2^{4H-1})\right)
\end{aligned}$$

For convenience we shift the entire process X_t by $-x_1$ so that the second point coincides with the origin. This makes it easier to work in polar coordinates. As a result, the conditional densities above become

$$\begin{aligned}
X_{2\Delta t} \mid (X_{\Delta t} = 0, X_0 = -x_1) &\sim \mathcal{N}\left((2^{2H-1} - 1)x_1, D\Delta t^{2H}(2^{2H+1} - 2^{4H-1})\right) \\
Y_{2\Delta t} \mid (Y_{\Delta t} = 0, Y_0 = 0) &\sim \mathcal{N}\left(0, D\Delta t^{2H}(2^{2H+1} - 2^{4H-1})\right)
\end{aligned}$$

Because the processes in x and y are independent, these variables have the joint distribution

$$f_{X_{2\Delta t},Y_{2\Delta t}}(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(\frac{-1}{2\sigma^2}\left((x-\overline{x})^2+y^2\right)\right)$$

where we have defined

$$\begin{aligned}\overline{x} &= (2^{2H-1}-1)x_1\\ \sigma^2 &= D\Delta t^{2H}(2^{2H+1}-2^{4H-1})\end{aligned}$$

Let R and θ be the polar coordinates of $X_{2\Delta t}$ and $Y_{2\Delta t}$, so that $X_{2\Delta t} = R\cos\theta$ and $Y_{2\Delta t} = R\sin\theta$. Then the joint density can be expressed

$$f_{R,\theta}(r,\theta) = \frac{1}{2\pi\sigma^2}\exp\left(\frac{-1}{2\sigma^2}\left((r\cos\theta-\overline{x})^2+r^2\sin^2\theta\right)\right)$$

Rearranging,

$$f_{R,\theta}(r,\theta) = \frac{1}{2\pi\sigma^2}\exp\left(\frac{-1}{2\sigma^2}(r-\overline{x}\cos\theta)^2\right)\exp\left(-\frac{\overline{x}^2\sin^2\theta}{2\sigma^2}\right)$$

The final step is to integrate out R to get the marginal distribution of θ. This requires the integral

$$\int_0^\infty r\exp\left(\frac{-1}{2\sigma^2}(r-\overline{x}\cos\theta)^2\right)\mathrm{d}r$$

This integral evaluates to

$$\sigma^2\exp\left(-\frac{\overline{x}^2\cos^2\theta}{2\sigma^2}\right)+\sqrt{2\pi\sigma^2}\overline{x}\cos(\theta)\Phi\left(\frac{\overline{x}\cos\theta}{\sqrt{\sigma^2}}\right)$$

where Φ is the unit Gaussian CDF:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-t^2/2}\mathrm{d}t$$

With this, we have the final angular distribution

$$f_\theta(\theta) = \frac{1}{2\pi}\exp\left(-\frac{\overline{x}^2}{2\sigma^2}\right)+\frac{\overline{x}\cos\theta}{\sqrt{2\pi\sigma^2}}\Phi\left(\frac{\overline{x}\cos\theta}{\sqrt{\sigma^2}}\right)\exp\left(-\frac{\overline{x}^2\sin^2\theta}{2\sigma^2}\right) \tag{3.29}$$

To summarize, we have derived the distribution for the angles formed by subsequent displacements of a fractional Brownian motion with Hurst parameter H and diffusion coefficient D, imaged at frame intervals of length Δt, with no localization

error, given that the first displacement has radial length x_1.

Fig. 3.9 shows several examples of the angular distribution for different values of the Hurst parameter, and table 3.1 illustrates the relationship between the angular distribution and the MSD for an FBM.

Hurst parameter	**angles**	**MSD**	**anomaly** α
$H < 1/2$	biased toward 180°	sublinear	$\alpha < 1$
$H = 1/2$	uniform	linear	$\alpha = 1$
$H > 1/2$	biased toward 0°	superlinear	$\alpha > 1$

Table 3.1: Distinct regimes of FBM as manifest in the angular distribution and MSD. The parameter α refers to the exponent in the common ad hoc equation $\mathrm{MSD}(t) \propto t^{\alpha}$.

The result is equally applicable for the modified diffusion coefficient $\overline{D}$ (Fig. 3.10), which is detailed in Appendix B. This is also a excellent demonstration of the role of the modified diffusion coefficient in separating the magnitude of the jumps from their correlation.

3.2.3 Fractional Brownian motion with localization error (FBME)

In real data, we never have processes corresponding to the fractional Brownian motion X_t considered above, due to the ubiquity of localization error. As we will see, localization error has strong consequences for the measurement of memory effects in diffusion. To examine these effects, here we consider a Gaussian process derived from FBM that incorporates a constant localization error term.

First, define the process $\overline{X}_t$ by adding some Gaussian noise to an FBM:

$$\overline{X}_t = X_t + N_t$$

Specifically, X_t is an FBM with Hurst parameter H and diffusion coefficient D, and N_t is white Gaussian noise such that

$$N_t \sim \mathcal{N}\left(0, \sigma_{\mathrm{loc}}^2\right)$$
$$\mathrm{Cov}\left[N_t, N_s\right] = \begin{cases} \sigma_{\mathrm{loc}}^2 & \text{if} \quad t = s \\ 0 & \text{otherwise} \end{cases}$$

The resulting stochastic process, $\overline{X}_t$, is another Gaussian process with the mean and covariance functions

$$\mathbb{E}\left[\overline{X}_t\right] = 0$$
$$\mathrm{Cov}\left[\overline{X}_t, \overline{X}_s\right] = \mathbb{E}\left[\overline{X}_t \overline{X}_s\right]$$

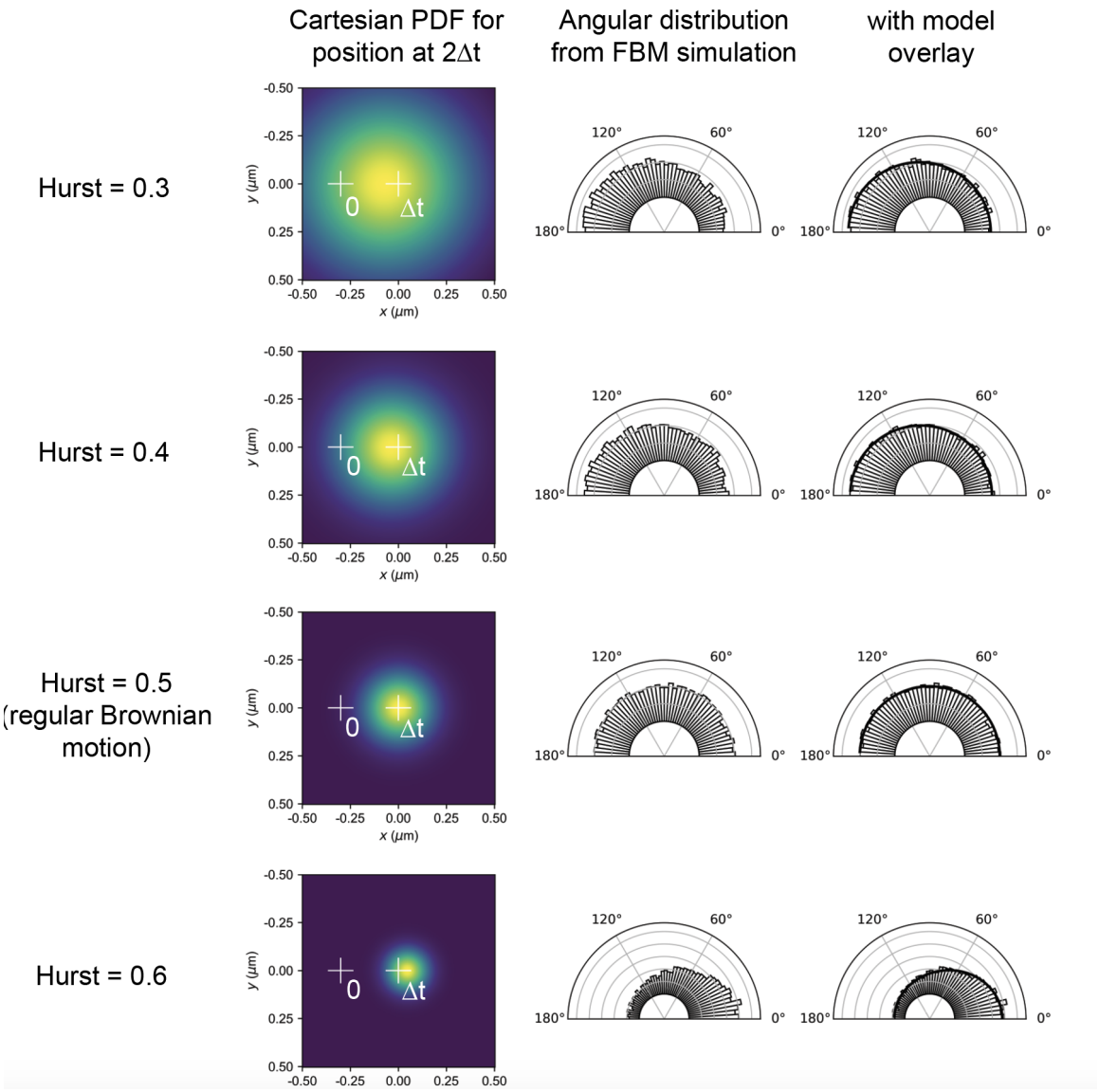

Figure 3.9: Some sample FBM angular distributions with the regular diffusion coefficient. FBMs were simulated and the outcome was compared with equation 3.29. The two crosshairs in the "Cartesian density" column correspond to the positions of the particle at time 0 and Δt. The length of the first jump is held constant at 300 nm in this case. The black line in the "model overlay" column corresponds to the prediction of equation 3.29. Noting the strong dependence between the variance of the Cartesian density and the Hurst parameter, compare this figure with Fig. 3.10.

$$textCov\left[\overline{X}_t, \overline{X}_s\right] = \begin{cases} 2Dt^{2H} + \sigma_{\text{loc}}^2 & \text{if} \quad t = s \\ D(t^{2H} + s^{2H} - |t-s|^{2H}) & \text{otherwise} \end{cases}$$

In effect, we have added a diagonal term to the covariance matrix. This changes the variance of each point's position without affecting the covariance between points. This process was considered recently by other authors [47].

Unlike for regular FBM, the first point $\overline{X}_0$ isn't necessarily zero due to the localization error. In reality we measure all of the points in the trajectory relative to the first point (localization error included, since its exact value in $\overline{X}_0$ is unknown to us).

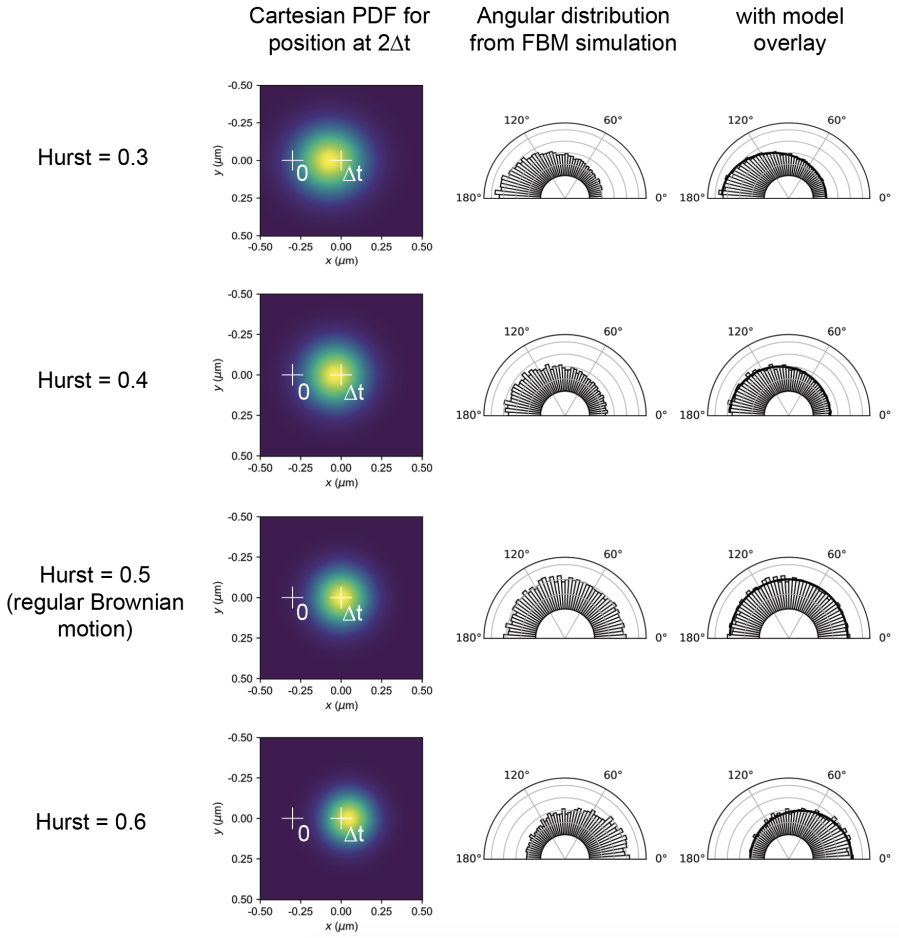

Figure 3.10: Some sample FBM angular distributions with the modified diffusion coefficient. FBMs were simulated according to simulation and equation 3.29, using the modified diffusion coefficient $\overline{D}$. The two crosshairs corresponds to the positions of the particle at time 0 and Δt. The length of the first jump is held constant at 300 nm. Note that using the modified diffusion coefficient prevents the Hurst parameter from exerting too much influence over the variance of the displacements, effectively decoupling the size of individual jumps from the correlation between jumps.

So we define a new stochastic process $\tilde{X}_t$ such that

$$\tilde{X}_t = \overline{X}_t - \overline{X}_0$$

Now $\tilde{X}_t$ is what we actually measure experimentally. This is another Gaussian process such that

$$\begin{aligned} \mathbb{E}\left[\tilde{X}_t\right] &= 0 \\ \text{Cov}\left[\tilde{X}_t, \tilde{X}_s\right] &= \mathbb{E}\left[\tilde{X}_t \tilde{X}_s\right] \\ &= \begin{cases} 0 & \text{if} \quad t = s = 0 \\ 2(Dt^{2H} + \sigma_{\text{loc}}^2) & \text{if} \quad t = s \neq 0 \\ D(t^{2H} + s^{2H} - |t-s|^{2H}) + \sigma_{\text{loc}}^2 & \text{if} \quad t \neq s \neq 0 \end{cases} \end{aligned} \tag{3.30}$$

Importantly, notice that the localization error of the first point enters into the covariance of $\tilde{X}_t$ and $\tilde{X}_s$ for $t > 0$. Strictly speaking, this means that even when $H = \frac{1}{2}$ (which, for an FBM, corresponds to regular Brownian motion), the process $\tilde{X}_t$ is not Markovian. As a result, in order to say anything about memory in diffusion, we have to be extremely aware of the effect of localization error on our measurements.

Elsewhere in this thesis, we'll refer to the Gaussian process $\tilde{X}_t$ just defined as *fractional Brownian motion with localization error*, or FBME.

In the sections that follow, we'll examine some of the properties of FBME, with special focus on the consequences for identifying and interpreting memory in SPT experiments.

Angular distribution for FBME

Since FBME is the closest model to what we actually measure on the microscope, it can give us insight into how localization error impacts our measurement. Here, we derive the distribution of angles formed by every three points in the trajectory of an FBME.

Suppose $\tilde{X}_t$ and $\tilde{Y}_t$ are independent FBMEs with Hurst parameter H and diffusion coefficient D that determine the x and y position of a diffusing particle, respectively. As in the previous case of FBM, we'll record the positions of these particles at three timepoints 0, Δt, and $2\Delta t$. For FBME, we always have $\tilde{X}_0 = \tilde{Y}_0 = 0$.

As before, we'll condition on $\tilde{X}_{\Delta t} = x_1$ and $\tilde{Y}_{\Delta t} = 0$, which we can rotate in the XY plane to obtain any radial displacement of size x_1. We can then apply the

conditioning property of Gaussian processes B.3 and shift by x_1 (exactly as in the case of FBMs) to derive the conditional densities

$$\tilde{X}_{2\Delta t} - x_1 \mid \left(\tilde{X}_{\Delta t} = x_1\right) \sim \mathcal{N}\left(\overline{x}, \sigma^2\right)$$
$$\tilde{Y}_{2\Delta t} \mid \left(\tilde{Y}_{\Delta t} = 0\right) \sim \mathcal{N}\left(0, \sigma^2\right)$$

where

$$\overline{x} = \left(\frac{D(2\Delta t)^{2H} + \sigma_{\text{loc}}^2}{2(D\Delta t^{2H} + \sigma_{\text{loc}}^2)} - 1\right) x_1$$
$$\sigma^2 = 2\left(D(2\Delta t)^{2H} + \sigma_{\text{loc}}^2\right) - \frac{\left(D(2\Delta t)^{2H} + \sigma_{\text{loc}}^2\right)^2}{2(D\Delta t^{2H} + \sigma_{\text{loc}}^2)}$$

This is quite ungainly, but it's worth examining the special case of $H = \frac{1}{2}$ before continuing, which can give some intuition about the role of localization error. In FBM with $H = 1/2$ we had $\overline{x} = 0$, meaning that the third point $\tilde{X}_{2\Delta t}$ on average landed on top of $\tilde{X}_{\Delta t}$. This is what one would expect for Brownian motion. But for an FBME we have $\overline{x} \leq 0$ (with equality holding only if the localization error is nonexistent). This is somewhat shocking. It means that, if we've seen two points in a Brownian trajectory with localization error, the mean of a future third point is *not* equal to the second point, but rather *lies between the first and second points*.

The magnitude of these deviations from martingale behavior depend on the ratio $D\Delta t/\sigma_{\text{loc}}^2$, and become very significant when dealing with slow-moving molecules.

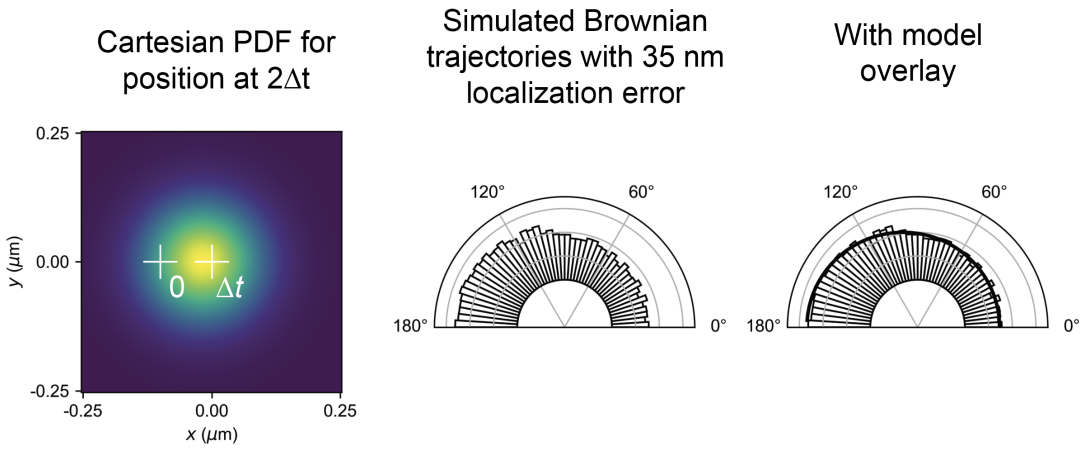

Figure 3.11: Demonstration of the effect of localization error on Brownian motion Relative positions in the trajectory were measured relative to the first point in a trajectory. In this case, $D = 0.2\ \mu\text{m}^2\ \text{s}^{-1}$ and the frame interval is 10 ms.

To see this, suppose that we have a Brownian motion with a diffusion coefficient $D = 0.2\ \mu\text{m}^2\ \text{s}^{-1}$ and localization error $\sigma_{\text{loc}} = 0.035$ nm. If we measure the particle's position at 10 ms frame intervals (as in 3.11), then the mean apparent jump between the first and second observations is about 100 nm. Taking $x_1 = 100$ nm, we have $\overline{x} \simeq -19$ nm. So despite that - on average - the *real* position of the particle does not change, the *apparent* position of the particle will undergo a mean shift of 19 nm back toward the first point in the trajectory, which represents $\sim 20\%$ of the distance of the first jump itself. Due to these results, we should be quite careful about analyzing memory in slow-moving molecules without taking into account the effect of localization error. The reader may note that this gives us some expectation about what to expect from the angular distributions of these slow-moving molecules.

Returning to the problem of angular distributions, we note that with the conditional densities above, the angular distribution is just 3.29 with the appropriate substitutions for $\overline{x}$ and σ^2. The general effect is to induce apparent subdiffusion, with a magnitude that depend strongly on the ratio $D\Delta t/\sigma^2_{\text{loc}}$.

Figure 3.12 examines the effects of localization error at several Hurst parameters for a slow-moving particle with $\overline{D} = 0.5\ \mu\text{m}^2\ \text{s}^{-1}$. Notably, the effect of the lo-

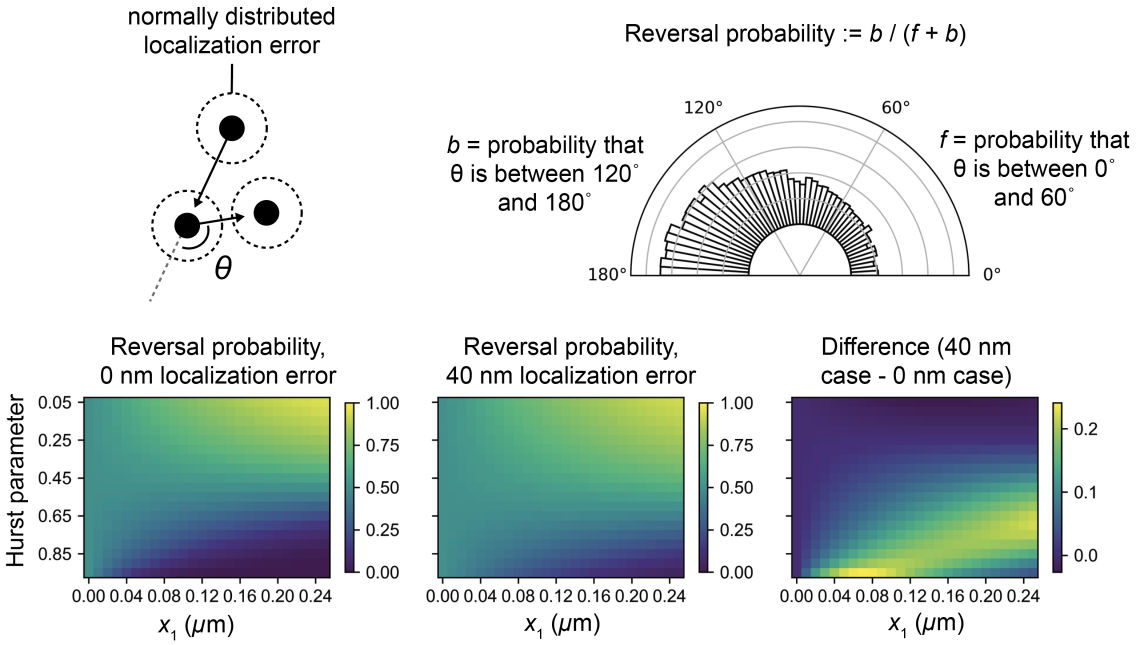

Figure 3.12: Effect of localization error on the reversal probability of FBMEs. FBMEs were simulated at various Hurst parameters for a particle with the modified diffusion coefficient $\overline{D} = 0.5\ \mu\text{m}^2\ \text{s}^{-1}$. Here we use 10 ms frame intervals.

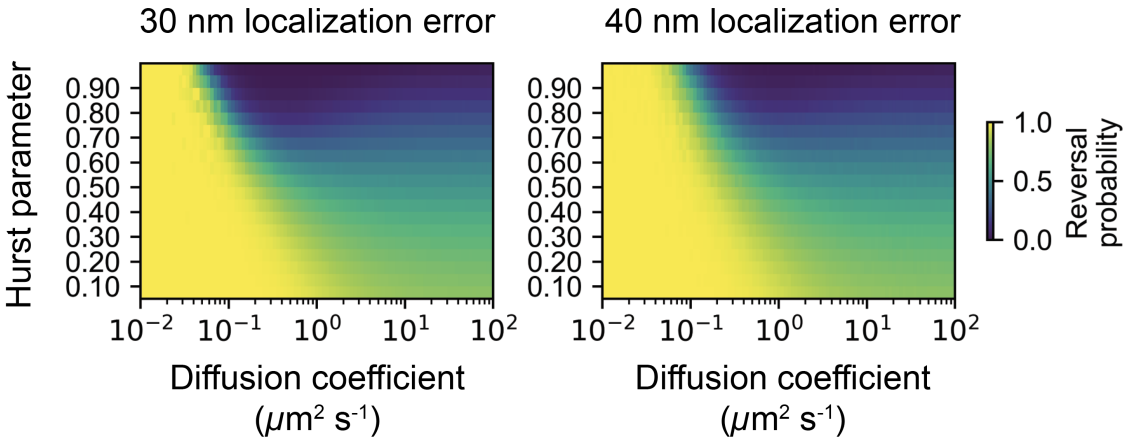

Figure 3.13: Reversal probabilities of FBMEs conditioned on long jumps. FBMEs were simulated as in Fig. 3.12, except the reversal probabilities were computed on all jumps greater than 0.16 μm rather than a specific jump length. The diffusion coefficient given here is the regular (non-modified) diffusion coefficient, and the frame intervals are 7.48 ms.

calization error is strongest for superdiffusion, essentially making directed motion impossible to detect by angular analysis. This is particularly important because the primary sources of directed motion in the cell - molecular motors including the myosin and kinesin networks - are intrinsically slow compared to diffusion.

However, the effect of localization error is decidedly less problematic for $H < 1/2$, which is the more common case observed inside cells. The main challenge is distinguishing these subdiffusive cases from genuine Brownian motion, which will always have a reversal probability greater than 0.5, despite being based on an underlying Markov process.

The problem can be mitigated by conditioning the angular distribution on jumps that are greater than some cutoff value - say, x_{min}. While the exact distribution of angles expected for an FBME is difficult to obtain analytically for this case, it is easy to evaluate with a Monte Carlo approach based on rejection sampling. The result of this procedure is shown in Fig. 3.13, which demonstrates the difficulty of inferring anomalous diffusion at low diffusion coefficients.

3.2.4 Increment process of FBMEs

We've seen that angular analysis of trajectories can be compromised by localization error, especially for slow-moving molecules. So does there exist another

nonparametric method to identify memory processes apart from the MSD and the angular distribution? In the following section, we will propose the increment covariance matrix as an alternative. To provide motivation and background for this proposal, here we develop a theory for the increment process of the FBMEs considered in the previous section.

Consider the process $\overline{X}_t = X_t + N_t$ considered in the previous section, where X_t is an FBM with Hurst parameter H and diffusion coefficient D and N_t is a Gaussian white noise process with $\mathrm{Cov}(N_t, N_s) = \sigma_{\mathrm{loc}}^2$ if $t = s$ and 0 otherwise. (This is equivalent to considering $\tilde{X}_t$; the extra localization error term will drop out.)

We'll imagine that we measure the position of this process at regular frame intervals $t_0 = 0$, $t_1 = \Delta t$, $t_2 = 2\Delta t$, and so on.

Define the FBME increment process as $Y_{i\Delta t} = \overline{X}_{i\Delta t} - \overline{X}_{(i-1)\Delta t}$ for $i = 1, 2, \ldots$. Then $Y_{i\Delta t}$ is another zero-mean Gaussian process such that

$$\begin{aligned} \mathbb{E}\left[Y_{k\Delta j}\right] &= 0 \\ \mathrm{Cov}\left(Y_{i\Delta t}, Y_{j\Delta t}\right) &= D\Delta t^{2H}\Big(|i-j+1|^{2H} + |i-j-1|^{2H} - 2\,|i-j|^{2H} \Big) \\ &\quad + 2\sigma_{\mathrm{loc}}^2 \mathbb{I}_{i=j} - \sigma_{\mathrm{loc}}^2 \mathbb{I}_{|i-j|=1} \end{aligned} \tag{3.31}$$

where $\mathbb{I}_{i=j}$ is the indicator function:

$$\mathbb{I}_{i=j} = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{otherwise} \end{cases}$$

Examining the specific case of regular Brownian motion for which $H = 1/2$, we have the increment covariance

$$\mathrm{Cov}\left(Y_{i\Delta t}, Y_{j\Delta t}\right) = \begin{cases} 2\left(D\Delta t + \sigma_{\mathrm{loc}}^2\right) & \text{if } i = j \\ -\sigma_{\mathrm{loc}}^2 & \text{if } |i-j| = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.32}$$

This highlights two important features of FBME. First, the increments are *stationary* as their covariance depends only on the magnitude of $|i-j|$ rather than the absolute values of i or j. Second, the role of localization error is to induce a negative correlation between subsequent jumps in a trajectory. The negative correlation is absent when comparing jumps that do not share a common detection.

3.2.5 Position and jump covariance matrices

Just as the MSD is the diagonal of the FBME covariance matrix (3.30), so is the angular distribution a representation of the diagonal and first off-diagonal positions of the increment covariance matrix (3.31). Neither function is free of the

effects of localization error, and in the case of the angular distribution, the role of localization error is actually fairly nonintuitive.

A simple alternative is to consider the full empirical covariance matrix. If $\mathbf{X}$ is an experimentally observed set of trajectories so that $\mathbf{X}_{ij}$ is the j^{th} position of the i^{th} trajectory, and so that the positions have been measured relative to the first point in each trajectory, then the sample position covariance is

$$\text{Cov}(\mathbf{X}) = \frac{1}{n-1}\mathbf{X}\mathbf{X}^T$$

where n is the number of trajectories, and the factor of $(n-1)^{-1}$ reflects the Bessel correction. The sample jump covariance can be obtained by the same procedure, except substituting $\Delta\mathbf{X}$, the matrix of jumps such that $(\Delta\mathbf{X})_{ij}$ is the j^{th} jump for the i^{th} trajectory.

When considering motion in 2D dimensions, we'll have two such matrices $\mathbf{X}$ and $\mathbf{Y}$ representing the positions in the x and y axes. It is then most convenient to consider the sample covariance

$$\text{Cov}\left(\{\mathbf{X}, \mathbf{Y}\}\right) = \frac{1}{2n-1}\left(\mathbf{X}\mathbf{X}^T + \mathbf{Y}\mathbf{Y}^T\right)$$

which assumes that the processes in the x and y dimensions are independent.

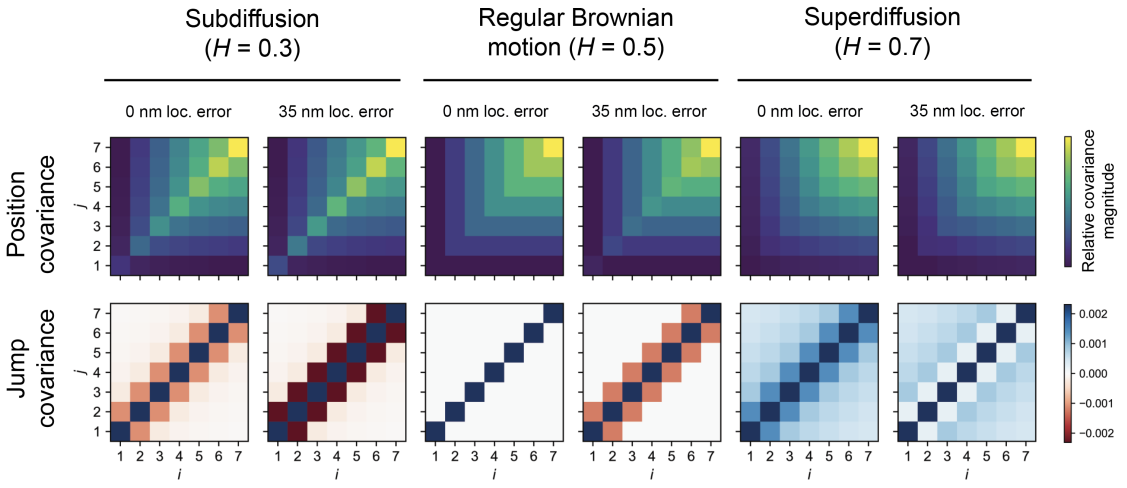

Figure 3.14: Visualization of the covariance matrices for several FBMEs. The modified diffusion coefficient was held constant at $\overline{D} = 0.5\ \mu\text{m s}^{-1}$ and the frame interval is $\Delta t = 0.00748$ ms. Note that while the jump covariances have all been colored according to the same scale, the position covariances have individually scaled colormaps for the purpose of illustration.

(We discuss the question of determining dependence in section 3.2.6 below.)

When computing the sample covariance on trajectories, two points are worth stressing:

1. To build the $n \times n$ covariance, all trajectories with fewer than n points (or $n+1$ points, if using the jump covariance) must be excluded from the matrix $\mathbf{X}$ (or $\Delta\mathbf{X}$).

2. To minimize the effects of localization error, which otherwise dominate the covariance matrix via their effect on the diagonal, it is a good idea to exclude immobile trajectories by imposing some lower limit on the trajectory mobility.

A common method to do the latter is simply to exclude trajectories for which the maximum likelihood estimate for the diffusion coefficient is below some lower limit. If Δx_i and Δy_i are the x and y displacements of a trajectory with length n, then the maximum likelihood estimate for its diffusion coefficient is

$$\hat{D}_{\text{mle}} = \frac{\sum_{i=1}^{n-1}(\Delta x_i^2 + \Delta y_i^2) - 4(n-1)\sigma_{\text{loc}}^2}{4(n-1)\Delta t}$$

where Δt is the frame interval and σ_{loc}^2 is the localization error. Alternatives include classifying trajectories into bound or free states via an HMM and then running the analysis on the free state, although this is not preferred because usually the HMM classification is highly parametric and can potentially inject difficult-to-track biases into the data.

Some covariance matrices for single-state regular Brownian motion are shown in Fig. 3.14. Notice in particular the zero off-diagonal terms in the jump covariance of a regular Brownian motion, and how the addition of localization error induces negative covariance between subsequent jumps in a trajectory.

Covariance matrices for several real spaSPT experiments are shown in 3.15. Notice that while the primary sources of variance for jumps are the diagonal terms (in a sense reflecting the "Markov" component of diffusion), the off-diagonal terms are significant in magnitude for RARA-HT and in particular H2B-HaloTag.

Covariance between x and y displacements

As a sanity check, we can also compute the covariance between the x and y displacements:

$$\text{Cov}\left(\Delta\mathbf{X}, \Delta\mathbf{Y}\right) = \frac{1}{n-1}\Delta\mathbf{X}\Delta\mathbf{Y}^T$$

where ΔX_{ij} is the *x* component of the j^{th} jump for the i^{th} trajectory, and likewise for ΔY_{ij}. As before, *n* is the total number of trajectories in the dataset.

Provided diffusion and localization error are isotropic, these sample covariances should be distributed around zero. Fig. 3.16 displays some of these "cross" covariances. The lack of consistent covariances between replicates suffices to demonstrate the isotropy of diffusion and localization error in these experiments.

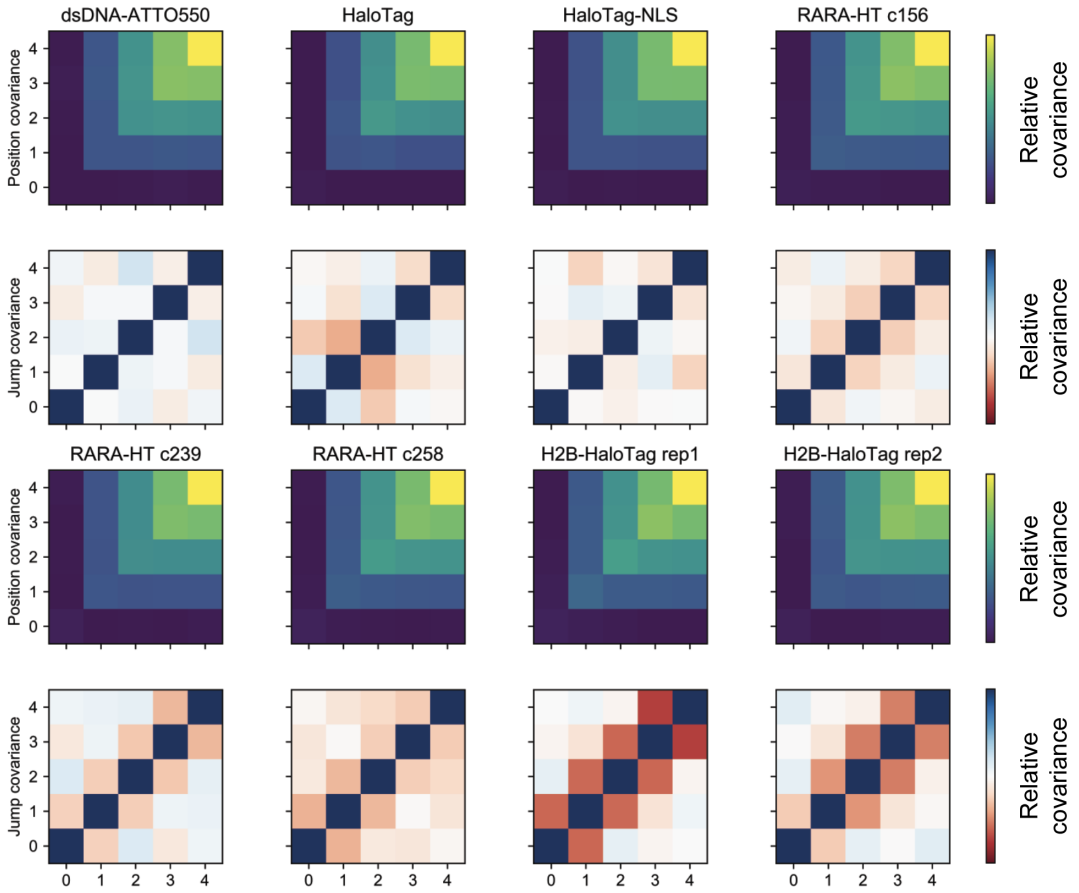

Figure 3.15: Covariance matrices computed on 7.48 ms tracking experiments with various biological samples. The colorscale of each subplot has been scaled individually. Note the stronger off-diagonal terms for RARA-HT and in particular H2B-HaloTag, reflecting a combination of localization error and subdiffusion. Both *x* and *y* displacements were considered in the computation of these matrices.

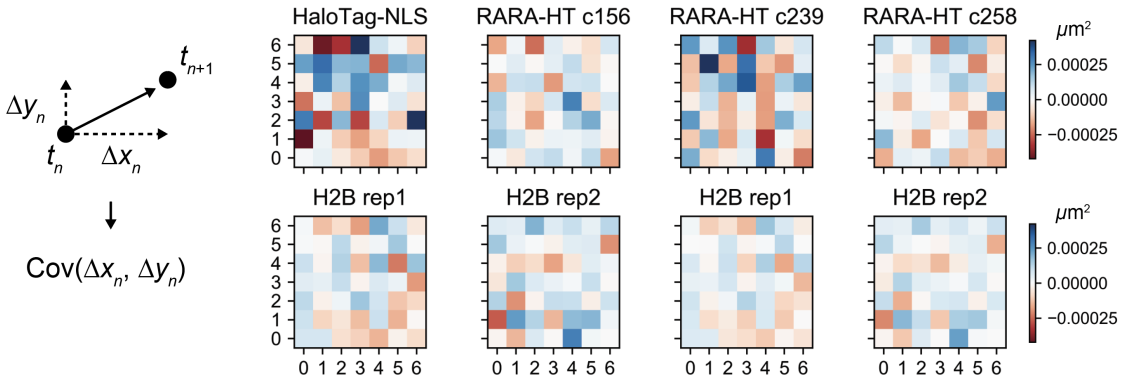

Figure 3.16: Cross covariance between the *x* and *y* components of jumps in experimentally observed trajectories in 7.48 ms tracking.

3.2.6 Separability of diffusion in the *x* and *y* dimensions

Nonzero covariance between the *x* and *y* components of the jumps (as considered in the previous section) would reflect something quite wrong with the experiment or image processing. However, zero covariance only implies *independence* if diffusion and localization error are both Gaussian processes. For various types of non-Gaussian diffusion, we may have zero covariance between the *x* and *y* components of the jumps, but these components may nonetheless be dependent processes. In this section, we examine this dependence through the lens of Levy flights, a popular model that exhibits this behavior.

A 2D diffusion process $\mathbf{R}$ with *x* coordinates $\mathbf{X}$ and *y* coordinates $\mathbf{Y}$ is *separable* in *x* and *y* if the joint PDF for its positions along each axis is factorable:

$$f_{\mathbf{R}}(\mathbf{x}, \mathbf{y}) = f_{\mathbf{X}}(\mathbf{x}) f_{\mathbf{Y}}(\mathbf{y})$$

FBMs and FBMEs, as Gaussian processes, are naturally separable. However, many other non-normal diffusion processes are inseparable. This includes all Levy flights except Brownian motion.

Let's examine the inseparability of Levy flights in more detail. Suppose that *X* and *Y* represent the *x* and *y* displacements for a single jump of a Levy flight with diffusion coefficient *D* and stability parameter α, and that $\mathbf{k} = (k_x, k_y)^T$ is the corresponding frequency vector. Then the process has the joint characteristic function

$$\phi_{X,Y}(\mathbf{k}, t) = \exp\left(-D\,|\mathbf{k}|^{\alpha}\, t\right)$$

Because $|\mathbf{k}|^{\alpha} = \left(k_x^2 + k_y^2\right)^{\frac{\alpha}{2}}$ only separates into a sum of k_x and k_y terms when $\alpha = 2$, and because separability in Fourier space implies separability in real space,

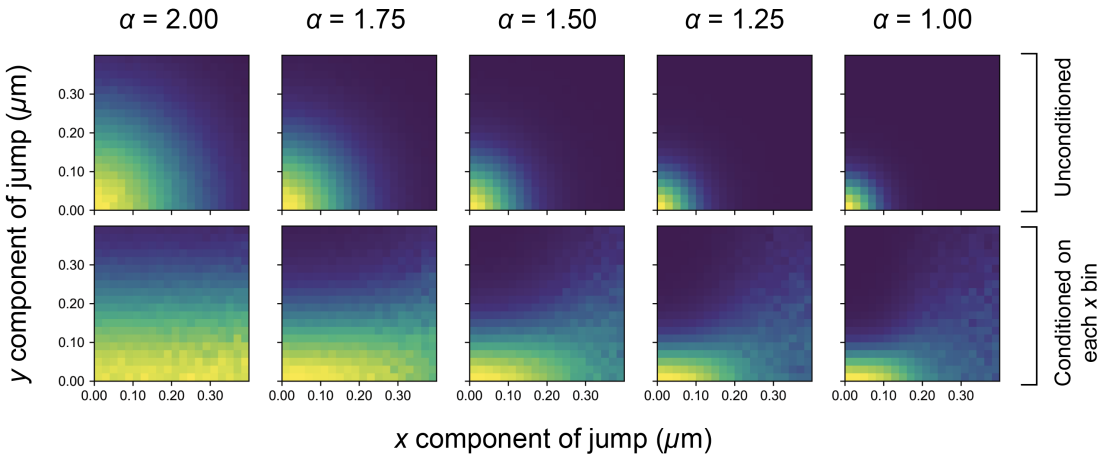

Figure 3.17: Mutual dependence of the magnitudes of *x* and *y* components of jumps for several types of Levy flights. The color scale reflects the relative probability of a jump falling into the respective bin. The lower row has been normalized over the bin corresponding to each *x* displacement. In these simulations, all Levy flights were given a diffusion coefficient of 2.0 μm^2 s^{-1} and were imaged at 7.48 ms intervals with 35 nm localization error in a 5 μm radius spherical nucleus with a 700 nm focal depth and a 13.4 Hz bleaching rate.

we can see that the PDF for the displacement will only be separable in the special case of regular Brownian motion.

The process remains isotropic and Markovian - at least if we ignore defocalization for the moment. What inseparability means here is that, when $\alpha < 2$, seeing the magnitude of a jump in the *x* dimension informs us about the magnitude of the corresponding jump in the *y* dimension. Specifically, a longer jump in *x* implies a longer jump in *y*.

We can use this insight to derive a simple test for separability: systematically vary the magnitude of the jumps in *x* and see how the magnitudes of the jumps in *y* respond. That is, we compute the conditional probability

$$\Pr\left(|\Delta y| \,\middle|\, |\Delta x|\right)$$

where Δx and Δy are the *x* and *y* components of a single jump in an spaSPT dataset.

Fig. 3.17 shows the result of this process for simulations of Levy flights with different stability parameters. Notice that only in the case of Brownian motion ($\alpha = 2$) is the displacement in the *y* dimension independent of the displacement in the *x*

dimension.

Fig. 3.18 shows the results of this analysis applied to several real datasets. In general, the *x* and *y* components of the displacements are not separable for real trajectories.

There is a major caveat associated with the separability approach to identifying anomalous diffusion: inseparability is a straightforward consequence of the presence of multiple diffusing states. Of the proteins shown in Fig. 3.18, only HaloTag and HaloTag-NLS could be reasonably approximated by a single diffusing state.

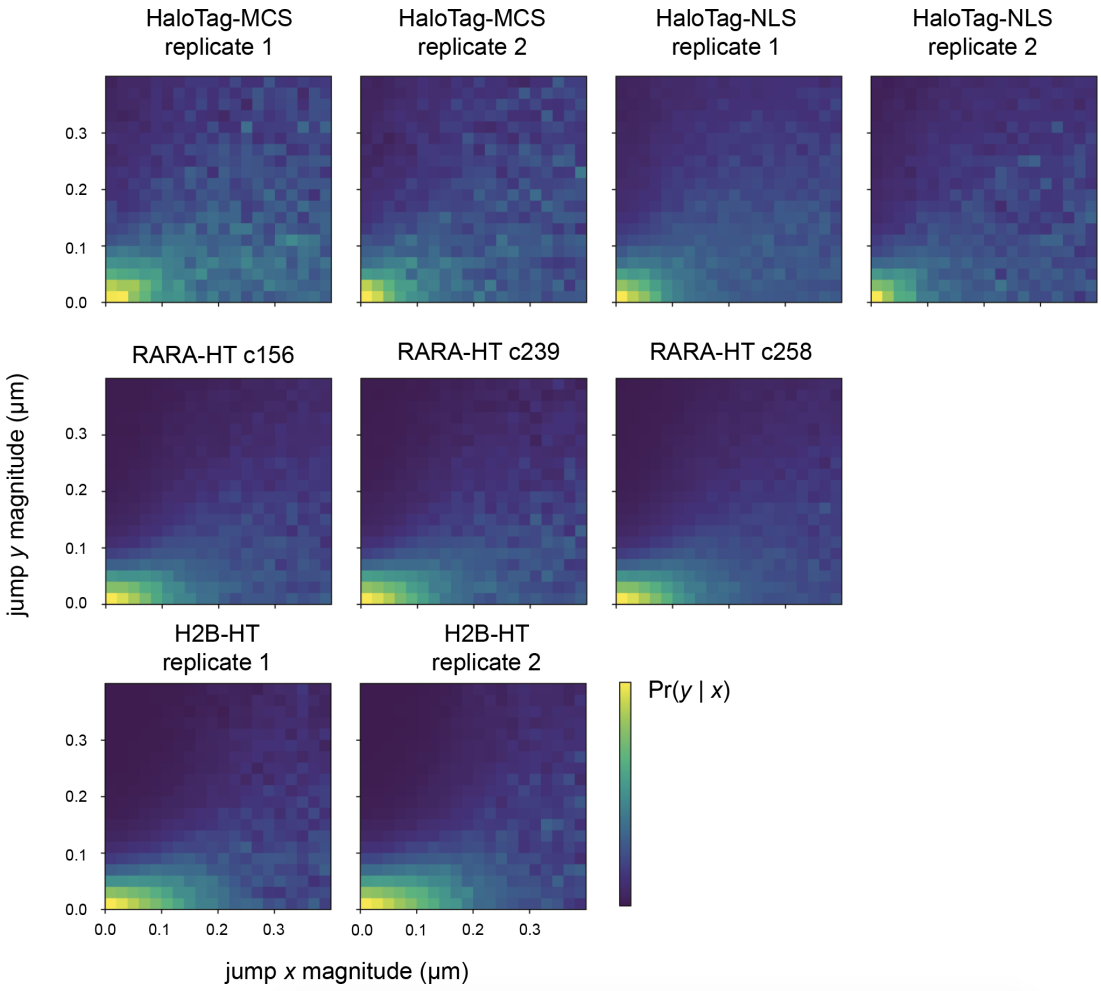

Figure 3.18: Mutual dependence of the magnitudes of *x* and *y* components of jumps for real trajectories collected in U2OS nuclei. All data were collected with the photoactivatable dye PAJFX549 at 7.48 frame intervals with 1.0 - 1.5 ms pulse widths and 1100 mW 561 nm laser power.

Determining whether inseparability is the result of anomalous diffusion or multiple diffusing states is a major challenge for this approach, and so it should be complemented by other approaches, particularly those based on identifying diffusing states in real data.

3.2.7 Aggregate likelihoods

One of the most central questions for the analysis of spaSPT data is how many diffusive states are present in a given dataset. One approach to identifying diffusive states is described in chapter 5 of this thesis, which relies on Bayesian methods to select between models. However, sometimes even this approach is too parametric and we need simpler methods.

A very simple approach is to simply plot the likelihood function for model parameters, aggregated across all trajectories in a dataset. The general idea is to evaluate the function

$$\Pr\left(\text{state } j\right) \approx \sum_{\text{trajectory } i} \Pr\left(\text{trajectory } i \text{ given state } j\right)$$

This approach has a justification in variational Bayesian statistics (see 5.4). Using this approach, we can systematically vary the model parameters to identify which are most probable, given an observed set of trajectories. This approach is quite useful for identifying multiple diffusing states in a nonparametric way.

In this section, first we provide a brief justification of the method, then illustrate its application to datasets using regular and fractional Brownian motion likelihood functions.

Principle

A full justification for the aggregate likelihood method in the context of variational Bayes is given in section 5.4. Here, we provide a short intuitive argument for the method.

Suppose we have a set of N trajectories, so that the i^{th} trajectory is X_i. Use $\mathbb{X}$ to denote the whole set of trajectories. (We use the "blackboard bold" typeface becauase $\mathbb{X}$ is often neither vector nor matrix, but a set of matrices of potentially variable shape. For instance, each X_i may represent the point coordinates of trajectory i.)

We'll imagine that our trajectories can inhabit one of K distinct diffusive states, each of which is parametrized by a parameter set θ_j ($j \in \{1, ..., K\}$). Let the vector

of all θ_j be $\boldsymbol{\theta}$. For regular diffusion, θ_j may simply be the regular diffusion coefficient. Along with the specification of the parameter sets, we also need a likelihood function, which we will write as

$$\begin{aligned} f_{X|\theta}(X_i \mid \theta_j) &= p(X_i \mid \theta_j) \\ &= \text{probability of trajectory } i \text{ given state parameters } j \end{aligned}$$

Now, we assume that each trajectory is generated from one of the K different states. Of course, for any single trajectory, we may have very little confidence which state that is. So we represent the state assignment as a random matrix $\mathbf{Z}$, where $Z_{ij} = 1$ if trajectory i is in state j and $Z_{ij} = 0$ otherwise. Since we are only allowing a trajectory to inhabit one state at a time, the rows of $\mathbf{Z}$ must sum to 1: $\sum_{j=1}^{K} Z_{ij} = 1$.

Assume that the true fractional occupancy of the different diffusive states is some vector $\boldsymbol{\tau}$, so that τ_j is the fractional occupancy of state j and $\sum_{j=1}^{K} \tau_j = 1$.

We don't have any reasonable belief as to what $\boldsymbol{\tau}$ should be before seeing any data, so assume that $p\left(\tau_j\right) = \text{constant}$ for all j. That is, we have no particular reason to favor one state over another. A natural way to represent this condition in Bayesian statistics is to choose the prior

$$\boldsymbol{\tau} \sim \text{Dirichlet}\left(n_0, ..., n_0\right)$$

In a full Bayesian framework, we would estimate the posterior distribution $p\left(\mathbf{Z}, \boldsymbol{\tau} \mid \mathbb{X}\right)$. However, in the aggregate likelihood method, we'll instead estimate $p(\boldsymbol{\tau} \mid \mathbf{Z}, \mathbb{X})$, the distribution over $\boldsymbol{\tau}$ given constant $\mathbb{X}$ and $\mathbf{Z}$. Of course, we don't know the state assignments $\mathbf{Z}$. So we'll make the mean field approximation

$$Z_{ij} \approx r_{ij} = \frac{f_{X|\theta}\left(X_i \mid \theta_j\right)}{\sum_{k=1}^{K} f_{X|\theta}\left(X_i \mid \theta_k\right)}$$

This actually corresponds to the posterior mean over $\mathbf{Z}$ given a uniform distribution for $\boldsymbol{\tau}$. (The full details are provided in 5.4.) Holding $\mathbf{Z}$ constant at $\mathbf{r}$, Bayes' theorem gives us

$$\begin{aligned} p(\boldsymbol{\tau} \mid \mathbf{Z}) &= \frac{p(\mathbf{Z} \mid \boldsymbol{\tau}) p(\boldsymbol{\tau})}{p(\mathbf{Z})} \\ &= \text{Dirichlet}\left(n_0 + \sum_{i=1}^{N} r_{i,1}, ..., n_0 + \sum_{i=1}^{N} r_{i,K}\right) \end{aligned}$$

For reasons outlined in chapter 4, we choose to weight the contribution of each trajectory to the posterior distribution by the number of jumps. If L_i is the number of jumps in trajectory i, then

$$p(\boldsymbol{\tau} \mid \mathbf{Z}) = \text{Dirichlet}\left(n_0 + \sum_{i=1}^{N} L_i r_{i,1}, ..., n_0 + \sum_{i=1}^{N} L_i r_{i,K}, ...,\right)$$

Finally, if we let the weight of the prior n_0 go to 0, then the mean of this posterior distribution is

$$\mathbb{E}\left[\tau_j \mid \mathbf{Z}\right] = \frac{n_j}{\sum_{k=1}^{K} n_k}$$

$$\text{where} \qquad n_j = \sum_{i=1}^{N} L_i r_{ij}$$

We use this posterior mean as the estimate for the occupancy of each state in the aggregate likelihood method. Intuitively, this is just the sum of the normalized likelihood functions for each trajectory. In chapter 5, we'll see that this represents a single step in the variational Bayes algorithm.

Regular Brownian motion likelihoods

Suppose that we have a trajectory in two dimensions with x-coordinates given by $\mathbf{X}_i$ and y-coordinates given by $\mathbf{Y}_i$, so that the k^{th} point in the trajectory is given by $(X_{i,k}, Y_{i,k})$. Further, imagine there are $L_i + 1$ total points in the trajectory.

As outlined in section 3.1, if the trajectory is a regular Brownian motion (RBM), its likelihood function can be approximated by

$$f_{X_i|\theta}\left(S_i \mid \theta_j\right) \approx \text{Gamma}\left(L_i, \frac{1}{4(\theta\Delta t + \sigma_{\text{loc}}^2)}\right) \tag{3.33}$$

where S_i is the sum of squared jumps for trajectory i:

$$S_i = \sum_{k=1}^{L_i} \left((X_{i,k+1} - X_{i,k})^2 + (Y_{i,k+1} - Y_{i,k})^2\right)$$

and θ_j is a diffusion coefficient.

Then the aggregated likelihood function for a set of trajectories $\mathbb{X}$ under the RBM likelihood function is

$$F(\theta_j) = \sum_{i=1}^{N} L_i f_{X_i|\theta}(S_i \mid \theta_j)$$

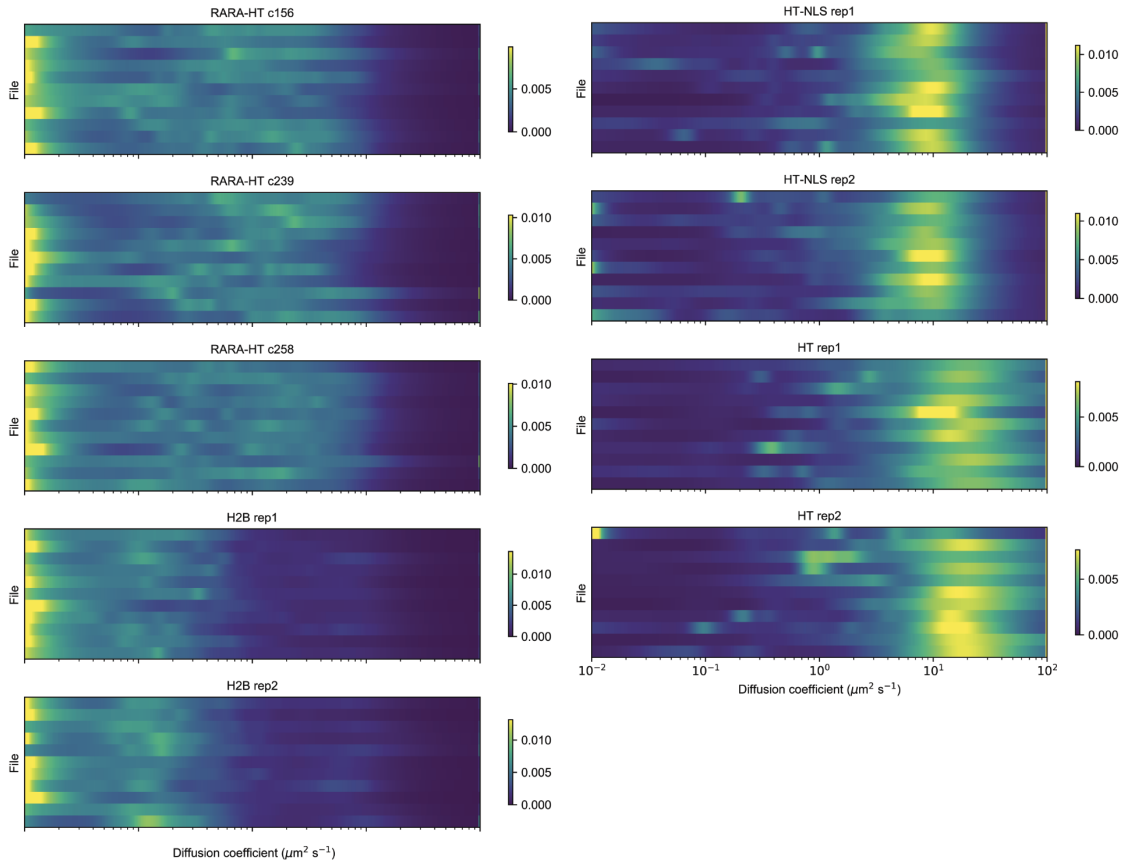

Figure 3.19: Aggregated likelihood function for regular Brownian motion. The labels for the subplots correspond to: *RARA-HT*: endogenously tagged retinoic acid receptor α-HaloTag in U2OS nuclei; *H2B-HT*: stably transfected histone H2B-HaloTag-SNAPf in U2OS nuclei; *HT*: transiently transfected HaloTag in U2OS nuclei; *HT-NLS*: transiently transfected HaloTag-3xNLS in U2OS nuclei. Cells were labeled with 100 nM PA-JFX549 for 30 min, then washed four times at 30 min for each wash. Imaging was performed with a 100X NA 1.49 objective under HiLo illumination with 7.48 ms frame intervals and 1 ms pulsed laser illumination. In these conditions, the approximate depth of field is 700 nm and the localization error is $\sim$ 35 nm. Each subplot represents a separate biological replicate, while each row of each subplot ("File") represents a separate nucleus.

Fig. 3.19 demonstrates the application of this function to some real trajectories. Importantly, we have not inferred the number or the diffusion coefficients of any diffusing states. Nevertheless, from this plot we can tell immediately that

1. RARA-HaloTag and H2B-HaloTag occupy a broader range of states than HaloTag or HaloTag-NLS.
2. While RARA-HaloTag and H2B-HaloTag have a substantial immobile fraction at the lower end of the distribution, HaloTag and HaloTag-NLS do not.

3. HaloTag diffuses much faster than HaloTag-NLS.

Some less obvious features are that

1. There is substantial variability in the immobile fraction of RARA-HaloTag and H2B-HaloTag between individual cells.
2. There is substantial variability in the slower-diffusing substates for HaloTag and HaloTag-NLS.
3. H2B-HaloTag has a low-occupation fast-diffusing state around $\sim 10\ \mu\text{m}^2\ \text{s}^{-1}$.

From these results, we see that a multi-state diffusion model may be merited for RARA-HT and H2B-HT, but not necessarily for HaloTag or HaloTag-NLS. In short, we can abstract a substantial amount of information from these plots without further analysis.

Fractional Brownian motion

The aggregated likelihood method is applicable to any trajectory likelihood function, not just the RBM likelihood 3.33. In later chapters, we'll see that a useful likelihood function to model memory effects in diffusion is the *fractional Brownian motion* likelihood

$$f_{\mathbf{X}|H,D}(\mathbf{x} \mid H, D) = \frac{\exp\left(-\frac{1}{2}\mathbf{x}^T C^{-1}\mathbf{x}\right)}{(2\pi)^{\frac{m}{2}} \det(C)^{1/2}}$$

where $\mathbf{X}$ is a one-dimensional trajectory, m is the number of spatial dmensions, and C is a covariance matrix with elements

$$C_{ij} = D\left((i\Delta t)^{2H} + (j\Delta t^{2H} - |(j-i)\Delta t|^{2H}\right)$$

Here, D is the diffusion coefficient, H is the *Hurst parameter*, and Δt is the frame interval. FBM treats diffusion in each dimension as independent, so the likelihood function for a 2D trajectory is just the product of the likelihoods for the trajectory along each axis. Exactly how localization error plays into the covariance matrix is explored in the next chapter. For now, it suffices to remark that when $H = 1/2$, we recover Brownian motion as a special case. For other values of H, the motion is no longer Markovian; memory effects enter into play. As a result, the FBM model is useful to describe molecules that exhibit subdiffusion or superdiffusion. As in the case of the RBM likelihood function, it is useful to evaluate the likelihood on a grid spaced logarithmically with respect to the diffusion coefficient to account for the higher dispersion of the faster states.

Fig. 3.20 shows the application of the FBM aggregate likelihood to simulated trajectories. Notice that, due to the short trajectory length in these simulated experiments, the dispersion of individual peaks tends to be quite broad, yet the true number of states can still be identified.

Fig. 3.21 applies the same method to experimental spaSPT datasets. Several features of the real dataset are worth mentioning:

- A small subpopulation of very high diffusion coefficient trajectories is apparent in all of the experiments. This is due to the contribution of tracking errors.
- H2B-HaloTag and RARA-HaloTag both have substantial populations at the lower end of the diffusion coefficient range, indicating the presence of an immobile population.
- H2B-HaloTag and RARA-HaloTag also both have slow-moving, highly subdiffusive states between 0.1 and 1.0 $\mu m^2\ s^{-1}$. From this plot alone, it can't be determined whether these states are due to bona fide subdiffusion or localization error.

Some of these features are annotated in Fig. 3.22. Altogether, the aggregate likelihood method provides a simple way to identify the number and approximate qualities of diffusing states in spaSPT data without subjecting them to model fitting.

3.2.8 Spot shape

Because the distribution of light presented by a mobile emitter is integrated over a finite time interval, this distribution represents the point spread function of the microscope convolved with the path of the particle. As a result, the spot itself can be used to parametrize diffusion independently of any correlative information (e.g. tracking) of spots between frames.

A simple method to parametrize spot shape is the Zernike transform [73], which is more frequently used to parametrize optical aberrations. The Zernike transform represents each spot as a linear combination of orthogonal functions on the unit disk called Zernike polynomials. By scaling the domain of the Zernike polynomials to a spatial extent appropriate for the molecule in question and extending the camera integration times to tens of milliseconds, it is possible to delineate the spatial compartments of the cell purely on the basis of the PSFs they present (Fig. 3.23).

We provide a software package, zzernike, for representing spots in spaSPT data as linear combinations of Zernike polynomials. The input is fully compatible with the output of the quot package.

3.2.9 Summary

Simple and nonparametric methods to identify the mode of motion are valuable for spaSPT data, especially since the workhorse in this area - the mean squared displacement - is rendered useless by the presence of multiple diffusing states with finite depth of field.

In this part, we have described several potential replacements for the MSD in order to characterize the mode of motion of a particle in a live cell spaSPT experiment. All are subject to shortcomings, but together they constitute a viable first pass at understanding the nature of a new spaSPT dataset. The merits and shortcomings of each approach are summarized in Table 3.2.

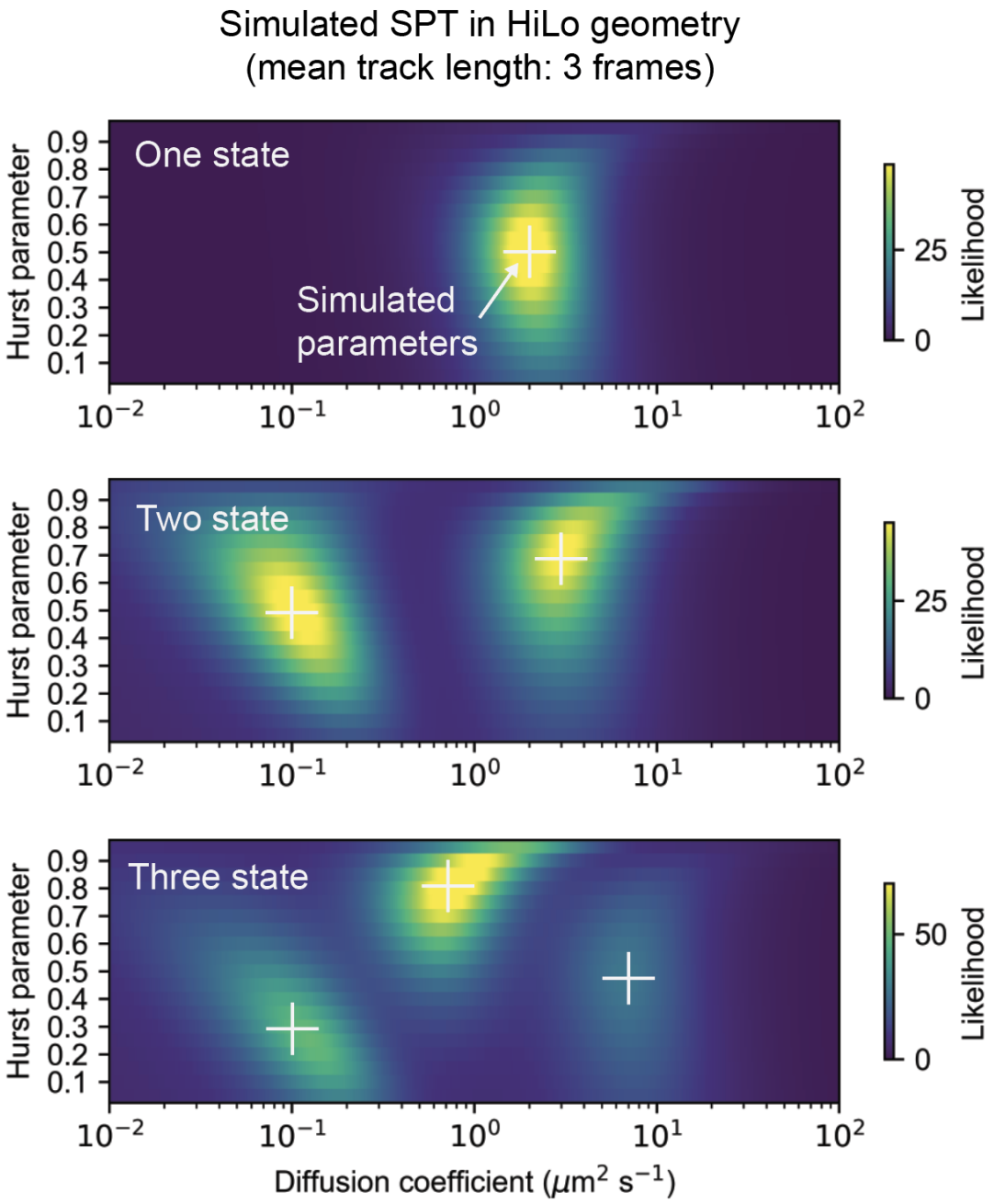

Figure 3.20: Aggregated likelihood function for fractional Brownian motion, evaluated on simulated trajectories in a HiLo geometry. White crosshairs indicate the simulated parameter sets, while the color maps are the aggregate likelihood function. In this simulation, we used a frame interval of 7.48 ms, 14 Hz bleaching rate, 5 μm radial nucleus, and a 700 nm focal depth. The mean trajectory length under these conditions is 3.5 frames, and about 10000 trajectories were used per subplot. The aggregated likelihood function for FBM was evaluated on a grid spaced logarithmically with respect to the diffusion coefficient.

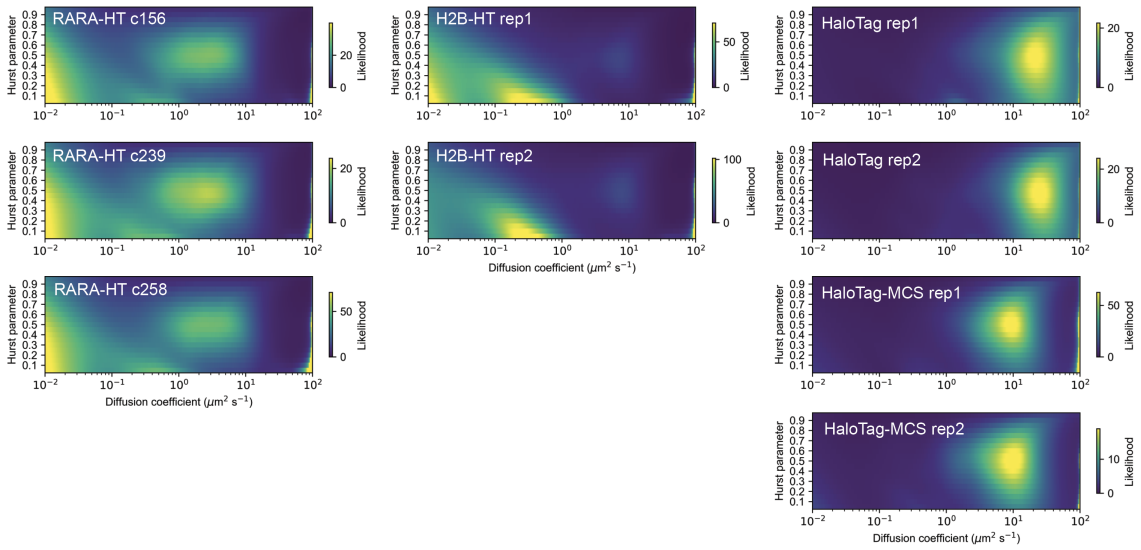

Figure 3.21: Aggregated likelihood function for fractional Brownian motion, evaluated on various experimental spaSPT datasets. All experiments were performed in U2OS nuclei as described in Fig. 3.19. Replicates indicate biological replicates, while the "c156", "c239", and "c258" next to the RARA-HT titles indicate independent knock-in clones. The absolute values of the likelihood function will depend on the size of the dataset, so the color maps have been scaled independently for each subplot. The aggregated likelihood function was evaluated a grid spaced logarithmically with respect to the diffusion coefficient.

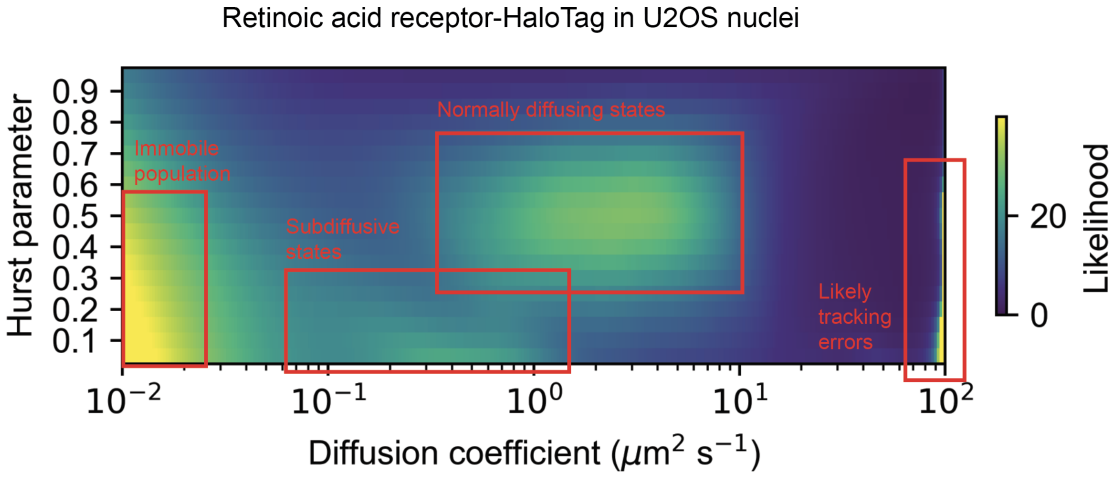

Figure 3.22: Aggregated likelihood function for fractional Brownian motion evaluated on retinoic acid receptor-HaloTag trajectories, with labeled features. All experiments were performed in U2OS nuclei as described in Fig. 3.19.

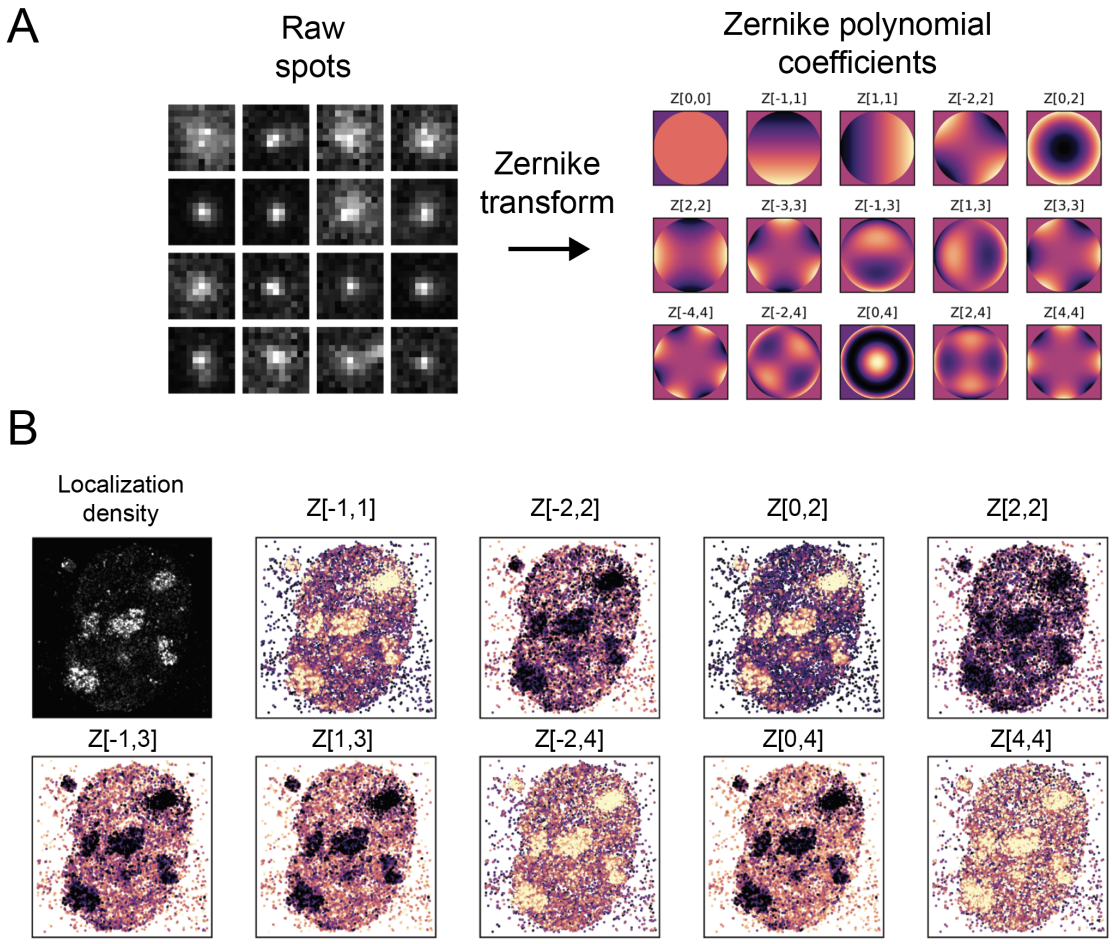

Figure 3.23: Using the Zernike transform to parametrize the mode of diffusion of nucleophosmin-HaloTag in different parts of the nucleus. (A) Schematic of the Zernike transform. A set of observed 2D spots are represented as a linear combination of Zernike modes on the unit disk. (B) Nucleophosmin-HaloTag-expressing U2OS cells were stained with PA-JFX549 and tracked with 30 ms exposures with continuous integration, rather than the stroboscopic schemes used elsewhere in this thesis. The color scale in each subplot reflects the value of the corresponding Zernike coefficient for each localization.

Method	Reports on	Advantages / shortcomings
MSD	Memory effects	Simple to compute and easy to identify localization error as the *y* intercept. But cannot distinguish between defocalization and subdiffusion in the presence of finite depth of field.
Jump angles	Memory effects	Independent of defocalization for separable diffusion processes, making it valuable as a complement to MSD. Sensitive to localization error, and effects of localization error are not always intuitive and easy to control.
Position covariance matrix	Memory effects, localization error	Contains all information about the diffusion for a Gaussian process, but is fairly nonintuitive for nonspecialists. Difficult to visually distinguish anomalous diffusion from normal diffusion without a statistical test.
Jump covariance matrix	Memory effects, localization error	More intuitive and makes the connection between subdiffusion and localization error clear. However, visualization is dominated by the magnitude of the diagonal term, which is dependent on the magnitude of the diffusion coefficient and localization error.
Conditional jump magnitude	Separability of diffusion in *x* and *y*	Simple to visually interpret and identifies inseparable diffusion processes. However, is sensitive to multiple diffusing states if these cannot be ruled out.
Aggregated likelihood	Presence of multiple diffusing states	Requires few trajectories, and can be used to get a rough idea of state occupancies. Sensitive to tracking errors.
Spot shape	Spatial variability in diffusion coefficient	Does not require tracking, so can be performed at higher densities. Requires longer integration time, slowing down acquisition. Probably cannot be used to measure non-Brownian modes of motion.

Table 3.2: Methods to identify modes of motion discussed in this chapter.

Chapter 4

Multiple diffusing states

Biological molecules characterized by a single diffusive state are the exception rather than the norm inside the cell. Local viscosities, transient interactions with the environment, and multimolecular complexes in which biomolecules participate often result in *mixtures of diffusive states*. For example, transcription factors present a distinct mode of diffusion when bound to the effective immobile scaffold of DNA than when diffusing freely through the nucleoplasm, or in complex with cofactors. As a result, different trajectories collected from the same cell can display remarkably varied modes of diffusion (Fig. 4.1).

A central goal in spaSPT analysis is to resolve these mixtures into individual components. This involves several related problems:

1. Identify the number of distinct states
2. Estimate the diffusive parameters for each state (*e.g.* diffusion coefficients and/or Hurst parameters)
3. Estimate the fractional occupancy of each state
4. Determine the most likely state(s) for each trajectory

In this chapter, we examine points (2) and (3), assuming that the number of diffusive states is known in advance. As we will see in the first section, these two points turn out to be linked due to the *defocalization problem*, which generates a dependence between the apparent fractional occupancy of a state and its diffusive parameters. Incorporating explicit knowledge of defocalization can remedy this problem and enable accurate state estimation for regular Brownian motion (RBM), fractional Brownian motion (FBM), and Levy flights. We then consider two approaches to solve the finite state estimation problem, one based on a maximum likelihood framework and one based on a least-squares framework.

At the end of this chapter, we return to points (1) and (4) above, reviewing existing approaches to infer the number of diffusive states from spaSPT data. This discussion lays the foundation for the next chapter, which generalizes the finite-state models considered here.

4.1 Defocalization

Consider an spaSPT experiment with two kinds of molecules - one slow-moving and one fast-moving. In the first frame, an equal number of both varieties are present in our field of view. We then track each of them to the next frame. How many jumps will we collect from each state?

If the microscope faithfully detects all fluorescent molecules in the cell, we would collect equal number of jumps from both states. However, spaSPT setups typically only resolve a thin slice of the cell. This shallow depth of field (or "focal depth") is a consequence of both the high NA objectives required to collect sufficient photons from each fluorophore as well as the inability to take *z*-stacks at

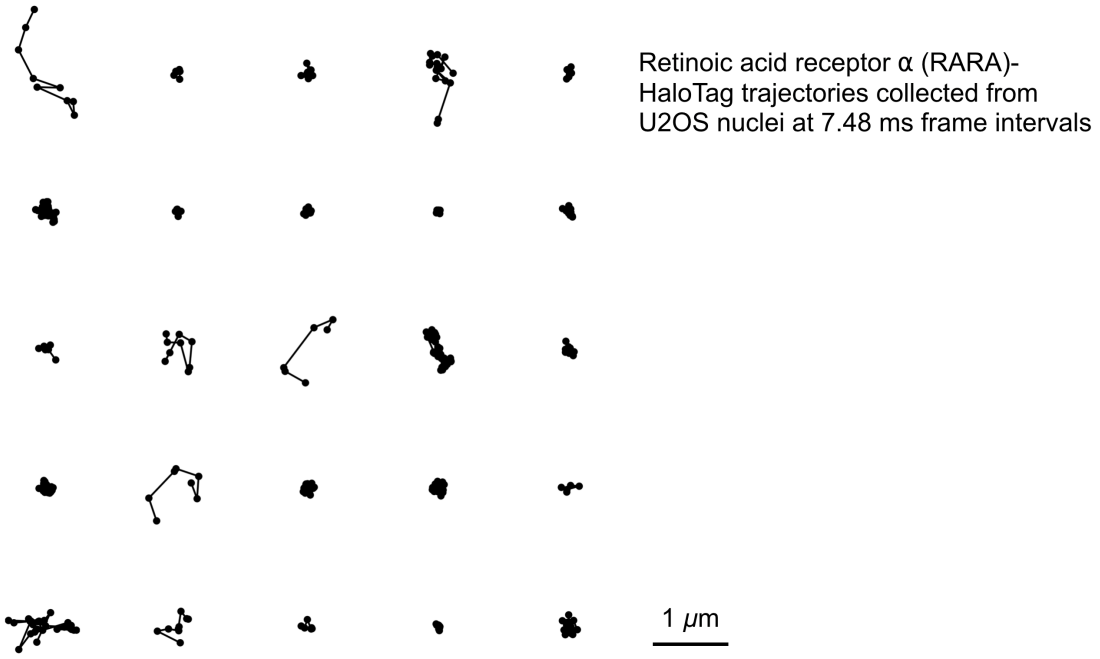

Figure 4.1: A set of randomly selected trajectories from an spaSPT dataset. Endogenous tagged retinoic acid receptor alpha (RARA)-HaloTag was labeled with PAJFX549 and tracked at 7.48 ms frame intervals with 1.5 ms pulse widths in live U2OS nuclei.

speeds requisite for tracking individual molecules. While multi-focal plane setups that would resolve many of these difficulties have been described [33], these are currently not widely available.

The result is that jumps are preferentially collected from the slow-moving states, which tends to remain within the focal volume (Fig. 4.2). The magnitude of this bias can be strong. For instance, if the slow-moving state has a diffusion coefficient of 0.1 μm^2 s^{-1} and the fast-moving state has a diffusion coefficient of 8.0 μm^2 s^{-1} - well within the range of experimentally observed variability - then 65% of jumps will be collected from the slow-moving state. In general, state estimators blind to the defocalization problem will systematically overestimate the occupancies of slow-moving states.

Defocalization was considered by Kues and Kubitscheck as a way to help measure the diffusion coefficient, since it induces a dependence between the diffusion coefficient and trajectory length [75]. Their approach was based on the assumption of absorbing boundaries at the edges of the detection slice. Because molecules can reenter the focal volume after leaving, this overestimates the defocalized fraction. Mazza implemented a similar correction for two-state regular Brownian motion models in spaSPT [59], and this approach was reimplemented by Hansen and Woringer [60]. Because these methods are based on Kues and Kubitscheck's

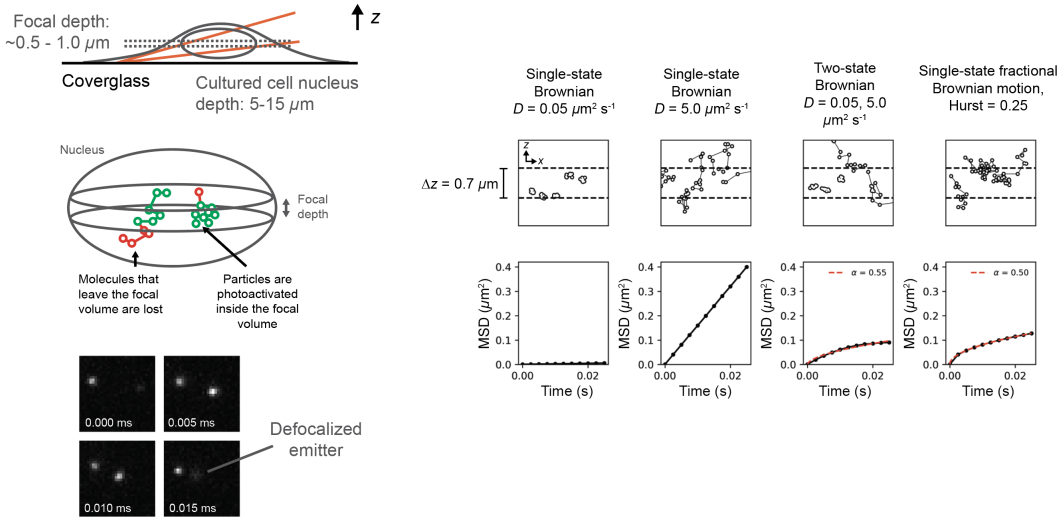

Figure 4.2: Schematic of the defocalization problem. Due to the finite depth of field ("focal depth") of most spaSPT setups, fast-moving molecules tend to contribute fewer observed jumps than bound molecules. This can lead to various misinterpretations, including systematically biased state estimation and apparent anomalous diffusion (sublinear MSDs).

model, which is fundamentally approximate, both Mazza and Hansen/Woringer's methods rely on correction factors derived from Monte Carlo simulations in order to accurately compute the fraction of defocalized molecules for a given diffusion coefficient. These simulations are performed in advance and then stored in a look-up table for subsequent model fitting. Unfortunately, because the number of Monte Carlo simulations quickly becomes unmanageable when considering all possible combinations of model parameters, focal depths, frame intervals, and gaps, the approach is unfeasible for models with more than a single diffusive parameter.

In this section, we describe a fast Fourier transform-based method to evaluate the defocalization function. This approach does not rely on Monte Carlo simulations or correctino factors, works for all focal depths and for any number of gaps, and extends easily to Markov processes beyond regular Brownian motion. In particular, we apply the method to Levy flights. We then consider a different method to address the defocalization problem for fractional Brownian motion, an important non-Markovian diffusion model. Integration of the defocalized correction with finite-state mixture model estimators are considered later in the chapter.

4.1.1 Computing defocalization for Markov processes

How do we account for the effect of defocalization on our data? In this section, we derive formulas to compute the fraction of particles that defocalize for several diffusion models. These formulas will subsequently be applied to state occupation measurements in spaSPT.

Consider the probability that a diffusing particle will leave the focal volume after some time $n\Delta t$, where Δt is the frame interval. Call this $p_{\text{defoc}}(n\Delta t \mid \boldsymbol{\theta})$. Here, $\boldsymbol{\theta}$ represents whatever parameters of the diffusion model are relevant to calculating this probability.

The form of p_{defoc} will depend on the specific diffusion model. As it turns out, if the diffusion process is Markovian - that is, if the jump between times 0 and Δt is independent of the jump between times Δt and $2\Delta t$ - then there is a simple algorithm to calculate p_{defoc}, which we describe here.

For now, assume we can observe particles if their z-position lies between $-\Delta z/2$ and $\Delta z/2$. If a particle is found within this range, it is assumed detected and tracked with probability 1.

Our trajectory starts out at some position $(X_0, Y_0, Z_0)^T$. We'll assume that the xy plane is infinite in extent and we can observe all of it. This is reasonable since the

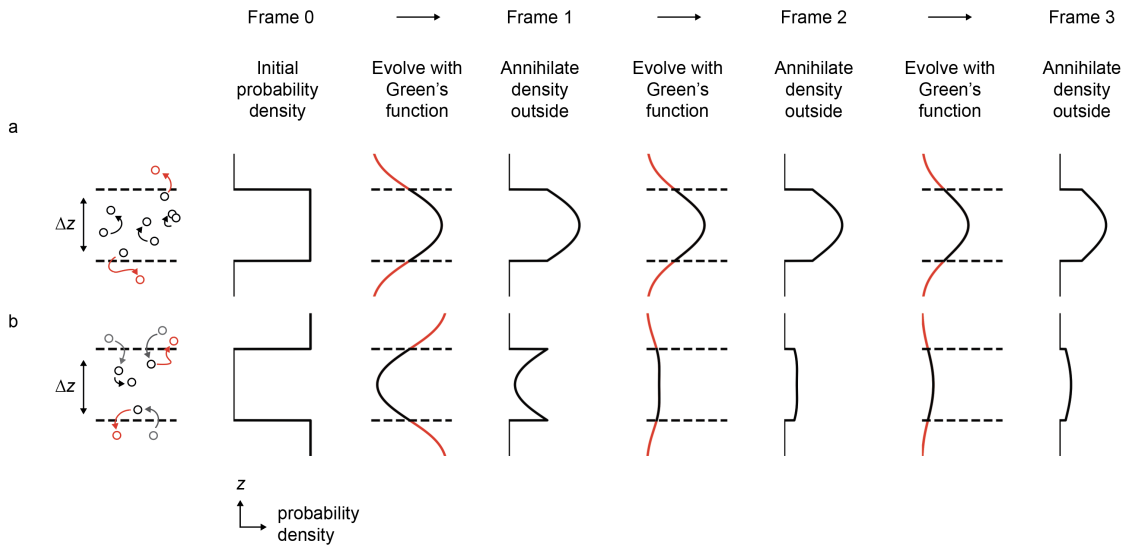

Figure 4.3: Schematic of an approach to calculate the fraction of observed trajectories that are contiguously observed in a finite-depth focal volume for gapless tracking. An initial density for detection is sequentially convolved with the Green's function for the diffusion model, then at each observation (corresponding to a laser pulse in stroboscopic tracking) the density outside the bounds of observation is set to zero.

average radius of a nucleus is $> 10\ \mu$m - a lot larger than experimentally observed jumps. So for simplicity we'll let $X_0 = Y_0 = 0$.

However, it matters where the particle starts in z because the depth of field is shallow. Particles that begin closer to the limits $\pm\Delta z/2$ have a greater chance to defocalize. So we'll imagine for the moment that our particle starts with uniform probability density between these limits:

$$Z_0 \sim \text{Uniform}\left(\frac{-\Delta z}{2}, \frac{\Delta z}{2}\right) \tag{4.1}$$

Use $f_{Z_0}(z_0)$ to denote the PDF corresponding to 4.1.

Let $f_{\Delta\mathbf{R}}(x, y, z) = f_{\Delta X, \Delta Y, \Delta Z}(x, y, z)$ be the PDF for a jump over some time interval Δt, so that position of the particle after one frame interval is $(\Delta X, \Delta Y, Z_0 + \Delta Z)^T$. If this jump density is radially symmetric, then there is some $f_{\Delta Z}(z)$ corresponding to the jump density along the z axis exclusively.

Then, using the convolution property C.6, the random variable $Z_0 + \Delta Z$ has the characteristic function

$$\phi_{Z_0+\Delta Z}(k) = \phi_{Z_0}(k)\phi_{\Delta Z}(k)$$

For instance, if our particle moves according to a Levy flight with stability parameter α and dispersion D and starts out with the density 4.1, then this is

$$\phi_{Z_0+\Delta Z}(k) = \text{sinc}\left(\frac{k\Delta z}{2\pi}\right)\exp\left(-Dt\,|k|^{\alpha}\right)$$

where

$$\text{sinc}\,(x) = \frac{\sin(\pi x)}{\pi x}$$

Then, using $\mathcal{F}_1$ to mean the one-dimensional Fourier transform, the fraction of molecules that remain within the focal volume after one frame interval is

$$\text{fraction inside focal volume after } \Delta t = \int_{-\Delta z/2}^{\Delta z/2} \mathcal{F}_1^{-1}\left[\phi_{Z_0}\phi_{\Delta Z}\right] \mathrm{d}z$$

Due to the Markov property, the jumps at later timepoints are independent of this first jump, and so the fraction of particles found within the focal depth after n frame intervals is

$$\text{fraction inside focal volume after } n\Delta t = \int_{-\Delta z/2}^{\Delta z/2} \mathcal{F}_1^{-1}\left[\phi_{Z_0}\phi_{\Delta Z}^{n}\right] \mathrm{d}z \tag{4.2}$$

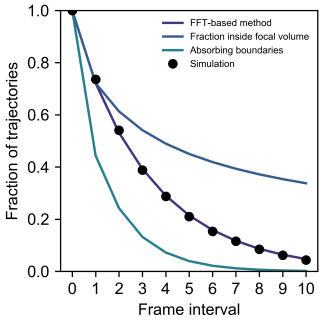

Figure 4.4: Comparison of Algorithm 7.1 with two other approximations to the defocalization function for regular Brownian motion. "FFT-based method" refers to Algorithm 4.1 in this text, "fraction inside focal volume" is the result of directly integrating the probability density inside the focal volume, and "absorbing boundaries" is the approximation used by Mazza, Hansen, and Woringer [59] [60]. The black dots reflect the results of simulation: trajectories with diffusion coefficient 2.0 μm^2 s^{-1} were photoactivated with uniform probability density in a 700 nm focal depth within a 10 μm spherical nucleus and were subsequently computationally imaged with 10 ms frame intervals.

However, this is not equal to the fraction of trajectories we actually observe in the focal volume after n frames. The reason is that many of these particles will make transits outside the focal volume for one or more of the intermediate frame intervals Δt, $2\Delta t$, ..., $(n-1)\Delta t$. Indeed, they may have immediately left the focal volume and returned only for a brief visit on $n\Delta t$. If we track without gaps, then these particles are lost - even if a particle is found outside the focal volume for a single frame interval, it will not be observed and the trajectory will be truncated. Equation 4.2 is always an underestimate.

In order to calculate the fraction of molecules that actually defocalize after n frame intervals, we need to remove the density present at each of the intermediate frame intervals. To do this, define the operator

$$\mathcal{K}_{\Delta z}[f](z) = \begin{cases} \mathcal{F}_1^{-1}\left[\phi_{\Delta z}\mathcal{F}_1(f)\right](z) & \text{if } z \in \left[-\frac{\Delta z}{2}, \frac{\Delta z}{2}\right] \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

The idea here is to convolve some initial density $f(z)$ with the Green's function for the diffusion model, then set all of the density outside the focal volume to zero. The approach is schematized in Fig. 4.3.

Then, assuming some initial density $f_{Z_0}(z)$, the experimentally observed profile in z after n frame intervals is

$$p_{\text{defoc}}(n\Delta t) = \int_{-\Delta z/2}^{\Delta z/2} \mathcal{K}_{\Delta z}^n \mathcal{F}_1\left[f_{Z_0}\right] \, \mathrm{d}z \tag{4.4}$$

where $\mathcal{K}_{\Delta z}^n$ means n sequential applications of the operator $\mathcal{K}_{\Delta z}$. Algorithm 4.1 describes this approach in detail.

Algorithm 4.1: Defocalized fraction of a Markov process after n frame intervals

Parameters:

- $f_Z(z, t)$, the model jump PDF over time t
- $g(z)$, a real-space transmission function. For instance, if we only observe particles between $\left[-\frac{\Delta z}{2}, \frac{\Delta z}{2}\right]$, then this might be

$$g(z) = \begin{cases} 1 & \text{if } z \in \left[-\frac{\Delta z}{2}, \frac{\Delta z}{2}\right] \\ 0 & \text{otherwise} \end{cases}$$

- Δz, the focal depth
- Δt, the experimental frame interval
- N, the number of frame intervals over which to compute defocalization

Precompute:

- $\phi_Z(k) = \mathcal{F}_1\left[f_Z(z, \Delta t)\right](k)$, the Green's function for the diffusion model over one frame interval

Algorithm:

Instantiate the result vector $\mathbf{v} \in \mathbb{R}^N$, where v_j is the fraction of molecules that have not defocalized at time $j\Delta t$.

Instantiate the density $p^{(0)}_{\text{curr}}(z) = f_0(z)$ with an appropriate numerical discretization.

For each $t = 1, 2, ..., N$:

1. Convolve the current density with the Green's function for the diffusion model, then apply the real-space transmission function:

$$p^{(t)}_{\text{curr}}(z) = g(z)\, \mathcal{F}_1^{-1}\left[\mathcal{F}_1\left[p^{(t-1)}_{\text{curr}}\right] \phi_Z\right](z)$$

2. Record $v_t = \int_{-\infty}^{\infty} p^{(t)}_{\text{curr}}(z) \mathrm{d}z$.

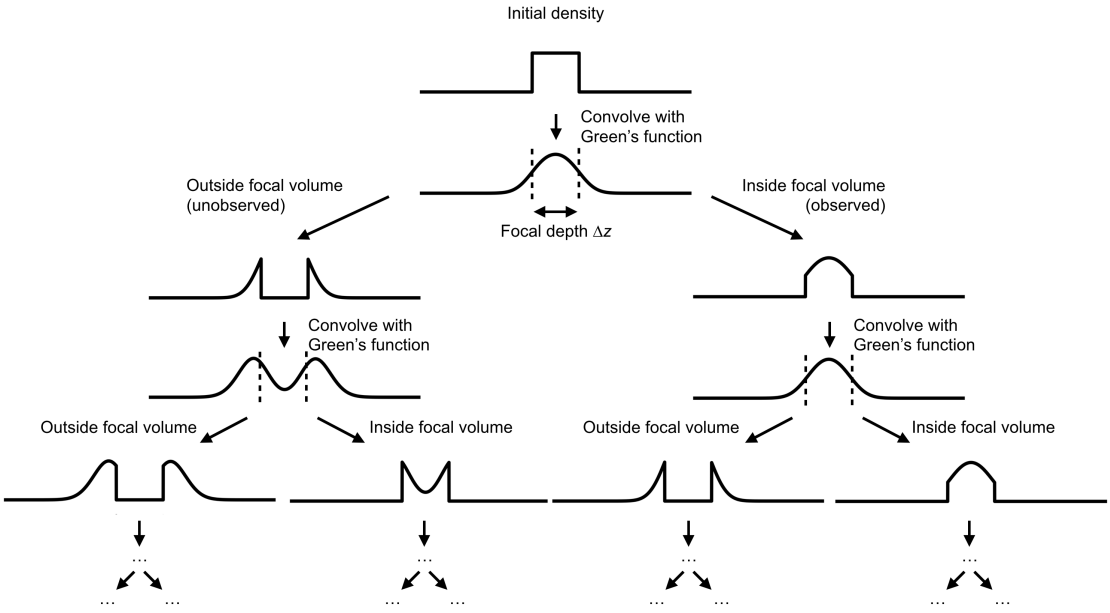

Figure 4.5: Schematic of the recursive approach to calculate defocalization with gaps. An initial probability density is sequentially evolved according to a diffusion model, then split into components that are inside and outside the focal volume. This is repeated at a recursion depth equal to the maximum number of frame intervals over which we want to calculate the defocalization correction. In this specific case, each line represents a 4 μm interval in the axial direction with a focal depth of $\Delta z = 700$ nm. The Green's function corresponds to Brownian motion with diffusion coefficient $D = 2.0$ μm^2 s^{-1} imaged at 10 ms frame intervals.

4.1.2 Defocalization with gaps

The method outlined in the previous section can be easily extended to tracking with gaps. Whereas in Algorithm 4.1 we set all of the probability density outside the focal volume to zero at each iteration, we can instead split off the probability density outside the focal volume and continue propagating it separately according to the Green's function for the diffusion model (Fig. 4.5).
Fig. 4.5 is a recursive approach, requiring a potentially large number of FFTs if the number of frames is high. However, by realizing that all of the density inside the focal volume can be aggregated into a single distribution, it can be modified to yield a fast iterative algorithm (Algorithm 4.2).

In this iterative algorithm, each $p^{(g)}_{\text{curr}}(z)$ represents the probability density for a particle that starts out in the focal volume at frame 0 and is subsequently outside the focal volume for g frames, up to some maximum tolerated number of gap frames n_{gaps}. When this is exceeded, the density that remains outside the focal

volume is lost forever. The algorithm works by taking the gapped density $p^{(g)}_{\text{curr}}(z)$, propagating it according to the diffusion model, then adding whatever density lands inside the focal volume back to the density inside the focal volume. This effectively sets the "gap count" for these trajectories back to zero. This whole cycle is repeated for however many frame intervals are relevant to the current analysis. The algorithm is linear in the product of the maximum tolerated gap count n_{gaps} and the number of frame intervals under consideration. In practice it is quite fast and amenable to iterative fitting algorithms.

Figure 4.6 compares Algorithm 4.2 with the result of simulated tracking data. Even for the fairly small number of trajectories in this experiment, the calculation is highly accurate. The effect of increasing the number of gaps is to decrease the proportion of trajectories that are lost to defocalization, since some of them return to the focal volume after a few gap frames. In particular, notice that the effect of gaps on tracking is to change the apparent state occupations by as much as 30%.

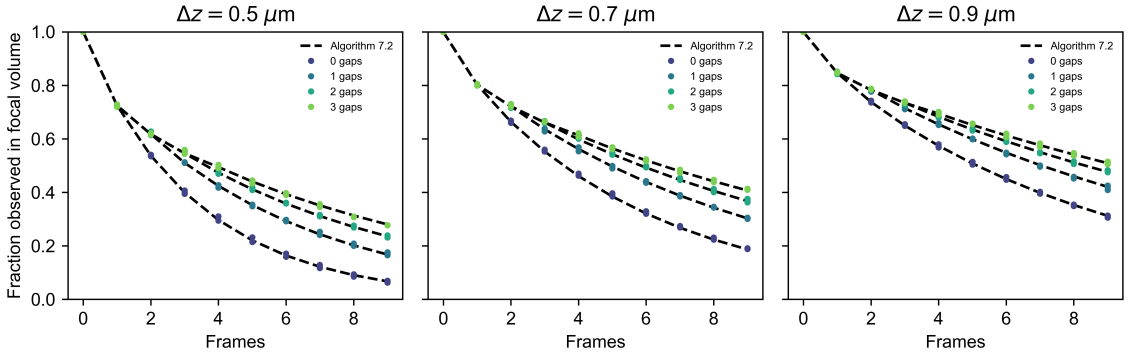

Figure 4.6: Comparison of the algorithm 4.2 with tracking simulations. Each subplot corresponds to a different focal depth Δz. At each focal depth, tracking was simulated with 10000 trajectories and the indicated number of gaps. Trajectories were initially photoactivated with uniform probability density in the interval $\left[-\frac{\Delta z}{2}, \frac{\Delta z}{2}\right]$. For all simulations, the mode of diffusion was regular Brownian motion with diffusion coefficient was held constant at $D = 2\ \mu\text{m}^2\ \text{s}^{-1}$ and frame interval $\Delta t = 0.00748$ ms.

Algorithm 4.2: Defocalized fraction of a Markov process for tracking with gaps

Parameters: $f_Z(z,t)$, the model jump PDF over time t; Δz, the focal depth; Δt, the frame interval; N, the number of frame intervals over which to compute defocalization; n_{gaps}, the maximum number of gaps allowed during tracking.
Precompute:

- $\phi_Z(k) = \mathcal{F}_1\left[f_Z(z,\Delta t)\right](k)$, the Green's function for the diffusion model over one frame interval

Algorithm:

Instantiate the result vector $\mathbf{v} \in \mathbb{R}^N$. v_j is the fraction of molecules that have not defocalized at time $j\Delta t$. Instantiate a set of buffers $p_{g,\text{curr}}(z)$ with $g = 0, 1, ..., n_{\text{gaps}}$, using an appropriate numerical discretization. For instance, a set of 1 nm bins from -2.0 μm to 2.0 μm works well for our purposes. Set the first buffer: $p_{0,\text{curr}}(z) = f_0(z)$, and the others to all 0. Also make two more auxiliary buffers $R(z)$ and $S(z)$ with the same discretization as $f_{g,\text{curr}}(z)$.

For each frame $t = 1, 2, ..., N$:

1. For each gap $g = n_{\text{gaps}}, n_{\text{gaps}} - 1, ..., 0$:
   (a) Set $S(z) := 0$ for all z.
   (b) Evolve the probability density by setting
   $$R(z) := \mathcal{F}_1^{-1}\left[\mathcal{F}_1\left[p_{g,\text{curr}}^{(t-1)}(z)\right]\phi_Z\right]$$
   (c) Take all of the probability density in $R(z)$ that lies within the focal volume and add it to the buffer $S(z)$:
   $$S(z) := S(z) + R(z) \text{ for all } z \in \left[-\frac{\Delta z}{2}, \frac{\Delta z}{2}\right]$$
   (d) Set $R(z) := 0$ for all $z \in \left[-\frac{\Delta z}{2}, \frac{\Delta z}{2}\right]$.
   (e) If $g < n_{\text{gaps}}$, then set $p_{(g+1),\text{curr}}^{t}(z) = R(z)$.
2. Set $p_{0,\text{curr}}^{(t)}(z) := S(z)$.
3. Integrate the fraction of particles remaining inside the focal volume at this frame interval:
   $$v_i := \int_{-\Delta z/2}^{\Delta z/2} p_{0,\text{curr}}^{(t)}(z)\,\mathrm{d}z$$

4.1.3 Non-uniform detection profiles in z

In our discussion of the defocalization problem so far, we have used two assumptions:

1. The particle starts out at a completely random position withinin the focal volume. That is, its axial position starts with uniform probability density between $-\Delta z/2$ and $\Delta z/2$.

2. At each frame, if the particle is inside the focal volume, it is detected and correctly tracked with probability 1.

In reality, detection may be harder at the edges of the focal volume than toward the center. We can accommodate a generalized detection profile as follows.

To relax assumption 1, we can substitute any initial profile $f_{Z_0}(z)$ in algorithms 4.1 and 4.2, instead of the uniform profile.

To relax assumption 2, examine algorithm 4.1. At each frame interval, we annihilated all probability density outside the focal volume by setting $p_{\text{curr}}(z) = 0$ for all $z \in \left[-\frac{\Delta z}{2}, \frac{\Delta z}{2}\right]$. This relied on the transmission function

$$g(z) = \begin{cases} 1 & \text{if } z \in \left[-\frac{\Delta z}{2}, \frac{\Delta z}{2}\right] \\ 0 & \text{otherwise} \end{cases}$$

Instead, we can filter the current density through any other $g(z)$ that reflects the probability to detect a particle at axial position z. The range of $g(z)$ should lie between 0 and 1, reflecting zero and unity probabilities of detection at that axial position.

To make the corresponding modification for the gapped tracking algorithm, notice that if $p_{\text{curr}}^{(t-1)}(z)$ is the probability density for the axial position at the $(t-1)^{\text{th}}$ frame, then $g(z)p_{\text{curr}}^{(t)}(z)$ is the probability density for the particles that are *observed* in the next frame and $(1-g(z))p_{\text{curr}}^{(t-1)}(z)$ is the probability density for particles that are *not observed* in the next frame. Then we can replace steps 1(c) and 1(d) in Algorithm 4.2 with the following:

$$\begin{aligned} &[\text{Step } 1(c)] && S(z) := S(z) + g(z)R(z)) \\ &[\text{Step } 1(d)] && R(z) := (1 - g(z))\, R(z) \end{aligned}$$

The rest of the algorithm remains the same. Since detection at the first frame of the trajectory and at subsequent frames of the trajectory has no real distinction

from an image processing perspective, it's usually appropriate to set the initial density as

$$f_{Z_0}(z) = \frac{g(z)}{\int_{-\infty}^{\infty} g(z)\mathrm{d}z}$$

4.1.4 Computing defocalization for fractional Brownian motion

Fractional Brownian motion (FBM) is an important non-Markovian category of diffusion. Algorithms 4.1 and 4.2 won't work for FBM because the behavior of an FBM trajectory will generally depend on its past. In other words, a single static Green's function cannot be used to evolve the probability density.

However, we can still account for defocalization by taking advantage of FBM's nature as a Gaussian process.

As outlined in Appendix B, the positions of a FBM along the z axis at a discrete set of time points $\mathbf{t} = (\Delta t, 2\Delta t, ..., n\Delta t)$ can be descibed as a multivariate normal random vector $\mathbf{Z}$ with covariance $\mathbf{C}$:

$$\begin{aligned}
\mathbf{Z} &\sim \mathcal{N}(\mathbf{0}, \mathbf{C}) \\
C_{ij} &= D\Delta t^{2H}\left(|i|^{2H} + |j|^{2H} - |i-j|^{2H}\right) \\
&= \overline{D}\Delta t\left(|i|^{2H} + |j|^{2H} - |i-j|^{2H}\right)
\end{aligned}$$

Here, $\overline{D}$ is the modified diffusion coefficient described in Appendix B. This means that the PDF for $\mathbf{Z}$ is

$$f_{\mathbf{Z}}(\mathbf{z}) = \frac{\exp\left(-\frac{1}{2}\mathbf{z}^T C^{-1}\mathbf{z}\right)}{(2\pi)^{\frac{n}{2}}\det(C)^{\frac{1}{2}}}$$

Suppose we have such an FBM that starts out at some initial position Z_0, which means that we shift the mean of the process above by Z_0. The probability that the particle is found within the focal volume at each of the discrete time points $\mathbf{t}$ is then

$$\Pr\left(\text{remain in focal volume} \mid \text{starting position } Z_0\right) = \int_{-\Delta z/2}^{\Delta z/2} \cdots \int f_{\mathbf{Z}}\left(\mathbf{z} - Z_0\right)\, \mathrm{d}\mathbf{z}$$

The integration is in a hyper-rectangular region defined by $z_j \in \left[-\frac{\Delta z}{2}, \frac{\Delta z}{2}\right]$ for all $z_j \in \mathbf{z}$. Then, supposing that $Z_0 \sim \text{Uniform}\left(-\frac{\Delta z}{2}, \frac{\Delta z}{2}\right)$, we can marginalize out Z_0

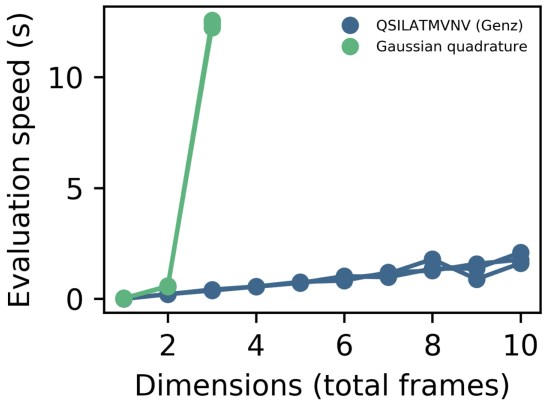

Figure 4.7: Comparison of the evaluation speeds for the FBM defocalization function. Computation of equation 4.5 was assessed for a Gaussian quadrature approach (from QUADPACK) against Alan Genz's QSILATMVNV method.

by taking

$$\begin{aligned} p_{\text{fbm defoc}}(\overline{D}\Delta t, H, \Delta z, n) &= \Pr\left(\text{remain in focal volume}\right) \\ &= \int_{-\Delta z/2}^{\Delta z/2} \mathrm{d}z_0 \int_{-\Delta z/2}^{\Delta z/2} \cdots \int \mathrm{d}\mathbf{z}\, f_{\mathbf{Z}}\left(\mathbf{z} - z_0\right) \end{aligned} \tag{4.5}$$

Here, we have emphasized the dependence on the parameters for the motion $\overline{D}$ and H, as well as the imaging parameters Δt, n, and Δz. Only the product $\overline{D}\Delta t$ is relevant for this integral, rather than $\overline{D}$ or Δt alone.

Unfortunately, this integral is intractable for traditional method such as Gaussian quadrature. However, there exist fast Monte Carlo methods for integrating these variables. In particular, the state of the art is represented by the methods of Alan Genz, particular the QSILATMVNV algorithm [74] which relies on an extremely efficient scheme to sample integrals over multivariate normal densities. The improvements of this method over Gaussian quadrature method is impressive (4.7).

In practice, we evaluate integral 4.5 at a discrete set of points in the space of $D\Delta t$, H, and Δz, then compute a cubic spline over this space for a quick approximation of the defocalization function at any other point in this space.

Fig. 4.8 compares the results of this approach against the results of simulation for FBMs with a variety of Hurst parameters and diffusion coefficients. Notice that both of these parameters have an effect on the defocalized fraction, even when using the modified diffusion coefficient. In particular, FBMs with higher

Hurst parameters tend to leave the focal volume sooner than FBMs with lower Hurst parameters.

4.2 Maximum likelihood estimators

In this section, we consider maximum likelihood estimators for the model parameters governing diffusive mixture models with a finite number of components. This discussion also serves to set up some constructions of mixture models that will prove useful in later sections, and demonstrates how the defocalization factors derived in the previous section can be used to improve state occupation estimates in spaSPT.

One advantage of maximum-likelihood estimators over the radial jump histogram-based estimators (considered later in this chapter) is that they remove several elements of choice in model fitting - including the number of frame intervals to consider, bounds on parameter values, and the parameters governing the binning of the jump distribution itself. But ML methods bring their own challenges too: the kind of mixture models we consider in this thesis have no closed-form maximum likelihood estimator. Instead we rely on iterative algorithms. The expectation-maximization (EM) algorithm is a classic choice for mixture models, and we develop it here in the context of spaSPT data.

This section has four parts. First, we state the problem of maximum likelihood

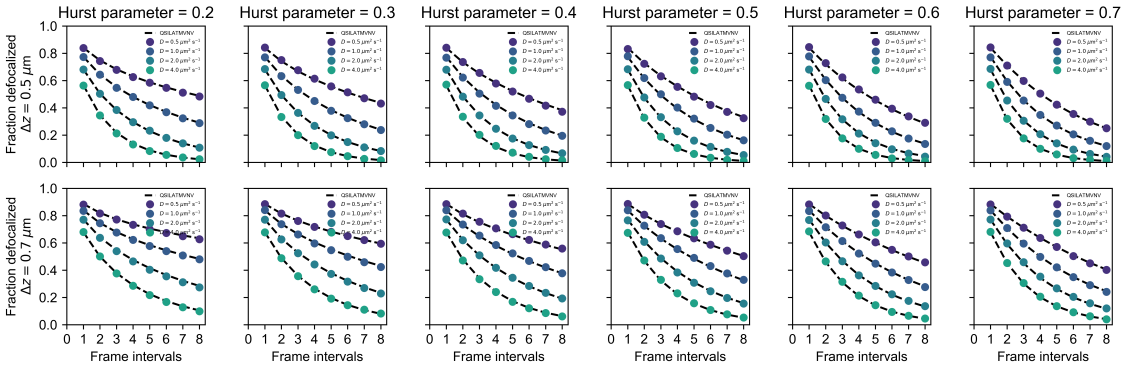

Figure 4.8: Comparison of simulation with the analytical defocalization equation 4.5 for various types of FBM. Each row of subplots corresponds to one of two focal depths - 0.5 or 0.7 μm - and each column corresponds to a different Hurst parameter. Dots correspond to the results of individual simulations, while dotted lines are the prediction of 4.5. All simulations were performed in a 10 μm spherical nucleus with specular reflections with 10 ms frame intervals. In these simulations, *D* refers to the modified diffusion coefficient described in Appendix B.

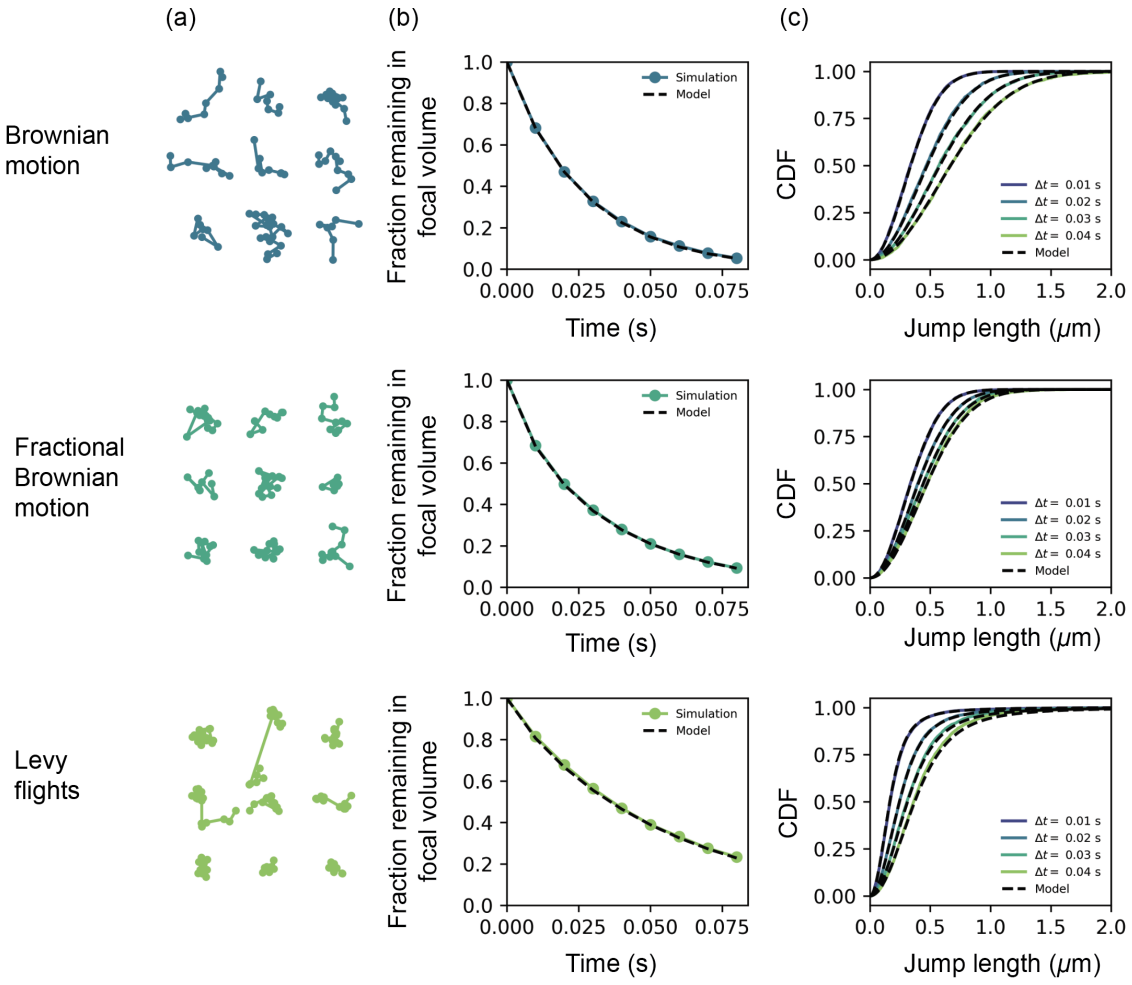

Figure 4.9: Defocalization functions and 2D radial jump CDFs for the three categories of diffusion considered in this thesis. For each category of diffusion, 10000 trajectories were photoactivated with uniform probability density in a 700 nm focal depth in a 10 μm spherical nucleus and positions were recorded with 35 nm localization error at 10 ms frame intervals. The simulation parameters for each model were as follows: Brownian motion, $D = 3.0\ \mu\text{m}^2\ \text{s}^{-1}$; FBM, $\overline{D} = 3.0\ \mu\text{m}^{-1}$, Hurst parameter 0.25; Levy flight, $D = 3.0\ \mu\text{m}^{1.5}\ \text{s}^{-1}$, $alpha = 1.5$.

inference for mixture models. Second, we discuss an EM routine for finding the ML parameters. Third, we discuss modifications of this algorithm to accommodate state biases arising from defocalization. Finally, we discuss instances of the algorithm applied to specific diffusive models.

4.2.1 Statement of the maximum likelihood problem

Given a diffusive model with parameter vector $\boldsymbol{\theta}$ and a set of observed trajectories $\mathbb{X}$, we seek the parameters that maximize a likelihood function:

$$\hat{\boldsymbol{\theta}}_{\text{mle}} = \underset{\boldsymbol{\theta}}{\text{argmax}}\, \mathcal{L}\left[\boldsymbol{\theta} \mid \mathbb{X}\right]$$

We use the "blackboard bold" symbol $\mathbb{X}$ to denote the set of trajectories because it is neither a vector nor a matrix, but a set of N matrices, each corresponding to a trajectory. We'll write these matrices as $\mathbf{X}_i \in \mathbb{X}$. Each matrix may be the spatial coordinates of a trajectory, the jump vectors of a trajectory, or some statistic calculated on the trajectory, and the different matrices $\mathbf{X}_i$ may not be the same size. Whatever is most convenient for the specific problem should be selected.

It's equivalent, and usually easier, to maximize $\log \mathcal{L}\left[\boldsymbol{\theta} \mid \mathbb{X}\right]$. If each trajectory $\mathbf{X}_i \in \mathbb{X}$ is independent of the others, then this log likelihood is

$$\log \mathcal{L}\left[\boldsymbol{\theta} \mid \mathbb{X}\right] = \sum_{i=1}^{N} \log \mathcal{L}\left[\boldsymbol{\theta} \mid \mathbf{X}_i\right]$$

In order to proceed, we need to choose the specific form for the likelihood of a mixture model. We'll construct the mixture model in the following way. First choose some fixed number of states K. Let $Z_i \in \{1, ..., K\}$ represent the diffusive state of trajectory i. We don't know Z_i; we only see the observed trajectory $\mathbf{X}_i$. Each state has some underlying proportion $\tau_j = \Pr\left(Z_i = j\right)$ in the mixture, which make up a vector $\boldsymbol{\tau}$. These proportions must satisfy $\sum_{j=1}^{K} \tau_j = 1$.

Let θ_j be the part of $\boldsymbol{\theta}$ that parametrizes the diffusive state j. For example, in a regular Brownian motion model, θ_j would be the diffusion coefficient for the j^{th} state. Together, the vectors $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$ are what we're trying to infer.

Use $f_{\mathbf{X}|Z}\left(\mathbf{x} \mid Z = j, \theta_j\right)$ to represent the probability density of observing a trajectory $\mathbf{X} = \mathbf{x}$, given that it inhabits state j and that the state is parametrized by θ_j. The specific form of $f_{\mathbf{X}|Z}$ will depend on the choice of diffusion model.

Then the likelihood function for a single trajectory, lacking knowledge of its state Z_i, can be written

$$\mathcal{L}\left[\boldsymbol{\theta}, \boldsymbol{\tau} \mid \mathbf{X}_i\right] = \Pr\left(\mathbf{X}_i \mid \boldsymbol{\theta}, \boldsymbol{\tau}\right) = \sum_{j=1}^{K} \tau_j\, f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j\right)$$

and the likelihood function for the whole dataset is

$$\mathcal{L}\left[\boldsymbol{\theta}, \boldsymbol{\tau} \mid \mathbb{X}\right] = \prod_{i=1}^{N} \sum_{j=1}^{K} \tau_j \, f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j\right) \tag{4.6}$$

Equation 4.6 is sometimes known as the "incomplete" log likelihood, because it represents the likelihood when have incomplete knowledge: we only know $\mathbb{X}$, not $\mathbf{Z}$. Unfortunately, it corresponds to the log likelihood

$$\log \mathcal{L}\left[\boldsymbol{\theta}, \boldsymbol{\tau} \mid \mathbb{X}\right] = \sum_{i=1}^{N} \log \left[\sum_{j=1}^{K} \tau_j \, f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j\right) \right]$$

Due to the sum within the logarithm, this is intractable for inference.

However, if we knew the vector of state assignments $\mathbf{Z}$, then the likelihood would become

$$\mathcal{L}\left[\boldsymbol{\theta}, \boldsymbol{\tau} \mid \mathbb{X}, \mathbf{Z}\right] = f_{\mathbb{X},\mathbf{Z}}\left(\mathbb{X}, \mathbf{Z} \mid \boldsymbol{\theta}, \boldsymbol{\tau}\right) = \prod_{i=1}^{N} \prod_{j=1}^{K} \left[\tau_j \, f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j\right)\right]^{\mathbb{I}_{Z_i=j}}$$

This is known as the "complete" likelihood, since it assumes knowledge of both the trajectories $\mathbb{X}$ and the state assignments $\mathbf{Z}$. $\mathbb{I}_{Z_i=j}$ is the indicator function, which is 1 if its argument is true and 0 if it is false. This reflects our knowledge of Z_i. Taking the logarithm, we have

$$\begin{aligned} \log \mathcal{L}\left[\boldsymbol{\theta}, \boldsymbol{\tau} \mid \mathbb{X}, \mathbf{Z}\right] &= \sum_{i=1}^{N} \sum_{j=1}^{K} \mathbb{I}_{Z_i=j} \left[\log \tau_j + \log f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j\right)\right] \\ &= \sum_{i=1}^{N} \left[\log \tau_{Z_i} + \log f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i, \theta_{Z_i}\right)\right] \end{aligned} \tag{4.7}$$

Unlike 4.6, equation 4.7 is quite tractable for inference. If for some reason we actually knew $\mathbf{Z}$, finding the maximum likelihood estimates $\hat{\boldsymbol{\theta}}_{\text{mle}}$ and $\hat{\boldsymbol{\tau}}_{\text{mle}}$ would be a cinch. In practice, numerous methods can still be used to maximize it while working around our ignorance about $\mathbf{Z}$. In the next part we examine one of these methods.

4.2.2 Expectation-maximization

The maximum likelihood solution to the finite mixture model can be obtained by an expectation-maximization (EM) routine. The idea here is that, even if we don't know the real vector of state assignments $\mathbf{Z}$, we can still work with a probability

distribution over $\mathbf{Z}$ that we'll refine along with our guess for the model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$. The algorithm iterates between two steps: (1) Determine the maximum-likelihood solutions for $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$ given the current distribution over $\mathbf{Z}$, and (2) recalculate the probability distribution over $\mathbf{Z}$ given the new estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$. A proof for the algorithm's convergence to the maximum likelihood solution can be found in the original EM paper [49]. An excellent general review is Chapter 9 in Bishop's book [48]. We provide an interpretation of the EM algorithm (including the merit function Q below) in section 5.4.

Call the parameter estimates for the t^{th} iteration $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\tau}^{(t)}$. Then define the merit function

$$\begin{aligned} Q\left(\boldsymbol{\theta}, \boldsymbol{\tau} \mid \boldsymbol{\theta}^{(t)}, \boldsymbol{\tau}^{(t)}\right) &= \mathbb{E}_{\mathbf{Z}|\mathbb{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\tau}^{(t)}}\left[\log \mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\tau} \mid \mathbb{X}, \mathbf{Z}\right)\right] \\ &= \mathbb{E}_{\mathbf{Z}|\mathbb{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\tau}^{(t)}}\left[\sum_{i=1}^{N}\left(\log \tau_{Z_i} + \log f_{\mathbf{X}|Z}(\mathbf{X}_i \mid Z_i, \theta_{Z_i})\right)\right] \\ &= \sum_{j=1}^{K}\sum_{i=1}^{N} \Pr\left(Z_i = j \mid \mathbf{X}_i, \boldsymbol{\theta}^{(t)}, \boldsymbol{\tau}^{(t)}\right)\left(\log \tau_{Z_i} + \log f_{\mathbf{X}|Z}(\mathbf{X}_i \mid Z_i, \theta_{Z_i})\right) \end{aligned} \tag{4.8}$$

Define the matrix $\mathbf{T}^{(t)} \in \mathbb{R}^{K\times N}$ such that

$$T_{ji}^{(t)} = \Pr\left(Z_i = j \mid \mathbf{X}_i, \boldsymbol{\theta}^{(t)}, \boldsymbol{\tau}^{(t)}\right) = \frac{\tau_j^{(t)} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j^{(t)}\right)}{\sum_{k=1}^{K} \tau_k^{(t)} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = k, \theta_k^{(t)}\right)} \tag{4.9}$$

Then the merit function becomes

$$Q\left(\boldsymbol{\theta}, \boldsymbol{\tau} \mid \boldsymbol{\theta}^{(t)}, \boldsymbol{\tau}^{(t)}\right) = \sum_{j=1}^{K}\sum_{i=1}^{N} T_{ji}^{(t)}\left(\log \tau_j + \log f_{\mathbf{X}|Z}(\mathbf{X}_i \mid Z_i = j, \theta_j)\right) \tag{4.10}$$

At each iteration, we seek

$$\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\tau}^{(t+1)} = \underset{\boldsymbol{\theta},\boldsymbol{\tau}}{\operatorname{argmax}}\, Q\left(\boldsymbol{\theta}, \boldsymbol{\tau} \mid \boldsymbol{\theta}^{(t)}, \boldsymbol{\tau}^{(t)}\right)$$

Since $\boldsymbol{\tau}$ and $\boldsymbol{\theta}$ appear in separate terms in equation 4.10, they can be maximized separately. The solution for $\boldsymbol{\tau}$ can be found with Lagrange multipliers as

$$\tau_j^{(t+1)} = \sum_{i=1}^{N} T_{ji}^{(t)} \tag{4.11}$$

In addition, each diffusive state appears as a separate term in 4.10. So they can also be maximized separately:

$$\theta_j^{(t+1)} = \underset{\theta_j}{\text{argmax}} \sum_{i=1}^{N} T_{ji}^{(t)} \log f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j\right) \tag{4.12}$$

The solution requires specification of the likelihood function $f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j\right)$. This depends on the type of diffusion (Brownian, FBM, etc.).

We highlight the important special case of regular Brownian motion.

4.2.3 Accounting for defocalization bias

The EM algorithm described in the previous section assumes that the frequency with which we observe each diffusive state reflects the true state occupations $\boldsymbol{\tau}$. This is not the case, for example, in situations with finite focal volume because some states escape the focal volume faster than others. If we have a trajectory in some state j, let η_j be the probability that our microscope observes this trajectory. In the general case, η_j is a function of the focal depth Δz, the frame interval Δt, the diffusion parameters θ_j, and the length of each trajectory L_j. Retaining the symbol $\boldsymbol{\tau}$ as the "true" state occupation vector, define the "observed" state occupation vector $\boldsymbol{\mu}$ such that

$$\begin{aligned} \mu_j &= \frac{\eta_j \tau_j}{\sum_{j=1}^{K} \eta_j \tau_j} \\ &\propto \eta_j \tau_j \end{aligned} \tag{4.13}$$

Assuming that $\mu_j \propto \eta_j \tau_j$ will be sufficient for our purposes, since we can impose normalization with Lagrange multipliers as shown below.

We can choose η_j to account for changes in the state occupation due to either defocalization or photobleaching. But since photobleaching affects all diffusive states equally, it factors out from the numerator and denominator in 4.13 and we're left with only the defocalization part.

Incorporating 4.13 into the complete log likelihood 4.7, we have

$$\log \mathcal{L}\left[\boldsymbol{\theta}, \boldsymbol{\tau} \mid \mathbb{X}, \mathbf{Z}\right] = \sum_{i=1}^{N} \left[\log \eta_j + \log \tau_{Z_i} + \log f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i, \theta_{Z_i}\right)\right] \tag{4.14}$$

Based on this, we can make the following modifications to the EM algorithm to account for defocalization biases. Let

$$T_{ji}^{(t)} = \Pr\left(Z_i = j \mid \mathbf{X}_i, \boldsymbol{\theta}^{(t)}, \boldsymbol{\tau}^{(t)}\right) = \frac{\eta_j \tau_j^{(t)} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j^{(t)}\right)}{\sum_{k=1}^{K} \eta_k \tau_k^{(t)} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = k, \theta_k^{(t)}\right)}$$

Then we have the merit function

$$Q\left(\boldsymbol{\theta}, \boldsymbol{\tau} \mid \boldsymbol{\theta}^{(t)}, \boldsymbol{\tau}^{(t)}\right) = \sum_{j=1}^{K} \sum_{i=1}^{N} T_{ji}^{(t)} \left(\log \eta_j + \log \tau_j + \log f_{\mathbf{X}|Z}(\mathbf{X}_i \mid Z_i = j, \theta_j)\right) \quad (4.15)$$

As before, at each iteration t, we seek

$$\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\tau}^{(t+1)} = \underset{\boldsymbol{\theta}, \boldsymbol{\tau}}{\operatorname{argmax}}\, Q\left(\boldsymbol{\theta}, \boldsymbol{\tau} \mid \boldsymbol{\theta}^{(t)}, \boldsymbol{\tau}^{(t)}\right)$$

To be exact, η_j is actually a function of θ_j. However, we will make the approximation that η_j can be treated as a constant for any given iteration. Then we can maximize the parts of $Q(\boldsymbol{\theta}, \boldsymbol{\tau})$ corresponding to $\boldsymbol{\tau}$ and $\boldsymbol{\theta}$ separately.

For the $\boldsymbol{\tau}$ part, we use Lagrangian multipliers. Impose the constraint $\sum_{j=1}^{K} \eta_j \tau_j = 1$. Then we seek the $\boldsymbol{\tau}$ that maximizes the Lagrangian

$$L(\boldsymbol{\tau}) = \sum_{j=1}^{K} \sum_{i=1}^{N} T_{ji}^{(t)} \left(\log \eta_j + \log \tau_j\right) - \lambda \sum_{j=1}^{N} \eta_j \tau_j$$

where λ is the Lagrange multiplier. Taking the first derivative and setting to zero, we have the solution

$$\tau_j^{(t+1)} = \frac{\sum_{i=1}^{N} T_{ji}^{(t)}}{\eta_j \sum_{k=1}^{K} \sum_{i=1}^{N} T_{ji}^{(t)}} \propto \frac{1}{\eta_j} \sum_{i=1}^{N} T_{ji}^{(t)} \quad (4.16)$$

This is the same solution with an additional bias factor η_j^{-1}. As before, finding the $\boldsymbol{\theta}$ is specific to each diffusion model.

4.2.4 Accounting for photobleaching

In a typical SPT experiment, fluorescent molecules only last for a few frames before bleaching.

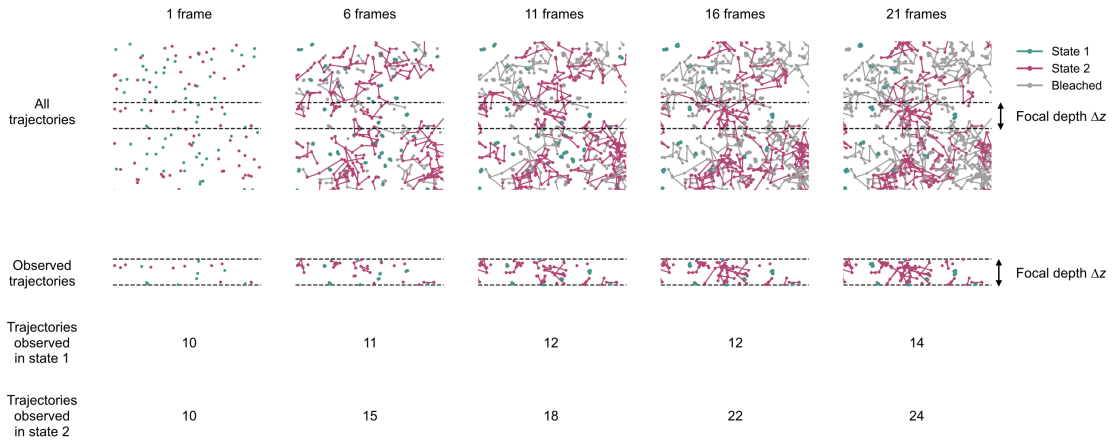

Figure 4.10: Some snapshots of a simulation demonstrating the state bias issue. The true occupations of state 1 and state 2 were each 50%. However, due to repeated reentry of the faster state into the focal volume, we count more apparent trajectories from state 1. In this simulation, the focal depth was 700 nm, the frame interval was 10 nm, the bleach rate was 20 Hz, and the diffusion coefficients corresponding to states 1 and 2 were $D_1 = 0.01\ \mu\text{m}^2\ \text{s}^{-1}$ and $D_2 = 2.0\ \mu\text{m}^2\ \text{s}^{-1}$ respectively.

This poses an issue for state occupations which echoes the defocalization discussions in the previous section. Imagine that we only have two classes of molecules - a completely immobile class and a highly mobile class. Fig. 4.10 illustrates this situation. All trajectories are subject to the same photobleaching rate, so they all eventually die out. However, we observe many more trajectories corresponding to state 2 than to state 1. The reason is that trajectories in state 1 - because they are slow - tend to contribute single, long trajectories within the focal volume. In contrast, trajectories in state 2 tend to transit multiple times through the focal volume before bleaching. As a result, we overcount the number of trajectories from state 2.

Any kind of state estimation based on counting the *fraction of trajectories* in different diffusive states will fail with finite focal volume.

Instead, we must weight our samples by *jump* rather than *trajectory*. This means that the fact that trajectories from state 2 reenter the focal volume multiple times doesn't matter. The only thing that we need to account for is the probability that a jump from either state lands inside the focal volume.

To accommodate this, we make the following simple modification to the EM algorithm. Instead of the matrix $\mathbf{T}$ defined above, instead weight each trajectory by the number of displacements. If L_i is the length of trajectory i in frames, then

define $\mathbf{T} \in \mathbb{R}^{K \times N}$ such that

$$T_{ji}^{(t)} = \Pr\left(Z_i = j \mid \mathbf{X}_i, \boldsymbol{\theta}^{(t)}, \boldsymbol{\tau}^{(t)}\right) = \frac{(L_i - 1)\eta_j \tau_j^{(t)} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j^{(t)}\right)}{\sum_{k=1}^{K} \eta_k \tau_k^{(t)} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = k, \theta_k^{(t)}\right)} \tag{4.17}$$

The remainder of the algorithm can be applied as described in the previous section.

4.2.5 EM algorithm applied to regular Brownian motion

Having developed the machinery for estimating state occupations, let's consider the specific case of regular Brownian motion (RBM). Suppose that we have a trajectory of length $n + 1$ frames measured in m spatial dimensions with diffusion coefficient θ_j and localization error σ_{loc}^2. Let the j^{th} spatial coordinate of this trajectory be $\boldsymbol{\xi}_j \in \mathbb{R}^m$. If we use the approximation 3.17, then the sum of squared displacements S_i has the distribution

$$S_i = \sum_{j=1}^{n} \left|\boldsymbol{\xi}_{j+1} - \boldsymbol{\xi}_j\right|^2 \sim \text{Gamma}\left(\frac{nm}{2}, \frac{1}{4(\theta_j \Delta t + \sigma_{\text{loc}}^2)}\right)$$

Let the vector of all such S_i be $\mathbf{S}$. This produces the EM merit function

$$Q\left(\boldsymbol{\theta}, \boldsymbol{\tau} \mid \boldsymbol{\theta}^{(t)}, \boldsymbol{\tau}^{(t)}\right) = \sum_{j=1}^{K} \sum_{i=1}^{N} T_{ji}^{(t)} \left[\log \eta_j + \log \tau_j - \frac{S_i}{4\left(\theta_j \Delta t + \sigma_{\text{loc}}^2\right)} - \frac{mn_i}{2} \log\left(4\left(\theta_j \Delta t + \sigma_{\text{loc}}^2\right)\right)\right]$$

Solving for $\partial Q / \partial \theta_j = 0$, we find

$$\theta_j^{(t+1)} = \frac{\sum_{i=1}^{N} T_{ji}^{(t)} S_i}{2m\Delta t \sum_{i=1}^{N} T_{ji}^{(t)} n_i} - \frac{\sigma_{\text{loc}}^2}{\Delta t}$$

Along with definition 4.17 and the state estimate update equation 4.16, this completes the EM algorithm for regular Brownian motion. This approach is summarized in Algorithm 4.3.

Algorithm 4.3: Maximum likelihood estimation for K regular Brownian diffusive states

Parameters: $\mathbb{X}$, a set of N experimentally observed trajectories in m spatial dimensions; K, the number of diffusive states; σ^2_{loc}, the localization error in μm^2; Δt, the frame interval; Δz, the focal depth.

Precompute: For each trajectory $i = 1, ..., N$, calculate the sum of squared radial jumps S_i and the number of jumps n_i.

Algorithm:

1. Choose some initial diffusion coefficients $\boldsymbol{\theta}^{(0)}$ and state occupations $\boldsymbol{\tau}^{(0)}$.
2. For each iteration $t = 0, 1, ...$:
   (a) Calculate the vector of state biases $\boldsymbol{\eta}$, given the current diffusion coefficients $\boldsymbol{\theta}^{(t)}$.
   (b) Calculate the state probabilities

$$T_{ji}^{(t)} = n_i \frac{\eta_j \tau_j^{(t)} f_{S|Z}\left(S_i \mid Z_i = j, \theta_j^{(t)}\right)}{\sum_{k=1}^{K} \eta_k \tau_k^{(t)} f_{S|Z}\left(S_i \mid Z_i = k, \theta_k^{(t)}\right)}$$

   Here, $f_{S|Z}(s|j, \theta_j)$ is the PDF corresponding to Gamma $\left(\frac{n_i m}{2}, \frac{1}{4(D\Delta t + \sigma^2_{\text{loc}})}\right)$.
   (c) Determine the new state occupations

$$\tau_j^{(t+1)} = \frac{\eta_j^{-1} \sum_{i=1}^{N} T_{ji}^{(t)}}{\sum_{k=1}^{K} \eta_k^{-1} \sum_{i=1}^{N} T_{ji}^{(t)}}$$

   (d) Determine the new diffusion coefficients

$$\boldsymbol{\theta}^{(t+1)} = \frac{\mathbf{TS}}{2m\Delta t \mathbf{Tn}} - \frac{\sigma^2_{\text{loc}}}{\Delta t}$$

   (e) If a convergence criterion on $\boldsymbol{\theta}$ and/or $\boldsymbol{\tau}$ is reached, terminate.

The most difficult part of the EM algorithm as currently stated is that we need an analytical solution for $\boldsymbol{\theta}^{(t+1)}$ at each step. Even in the case of regular Brownian motion, we need to introduce an approximation (neglecting the correlation between

subsequent jumps due to localization error) in order to produce this solution.

To avoid analytical solutions for the model parameters at each step, we can either resort to MCMC methods (such as Gibbs sampling) or jump length histogram estimators.

4.3 Gibbs sampling

While the EM algorithm is powerful, it requires analytical solutions for the parameters that maximize the conditional likelihood at each step. These are often hard to obtain. Another challenge is the one inherent to maximum likelihood methods - EM only generates a single point estimate of the model parameters. If many other sets of parameters describe the data equally well, we will not learn this by EM (or radial jump histograms, for that matter).

A potential solution to both of these problems is to use Markov chain Monte Carlo (MCMC) methods. Because MCMC methods only require evaluation of the likelihood rather than its derivatives and do not require closed-form solutions at each step, they can be more easily applied to complex diffusion models. And because they return a distribution over model parameters rather than point estimates, they can also be used to judge whether one set of parameters describes the data better than others. Nevertheless, MCMC methods have their own drawbacks. These include the *identifiability problem*, the method's inherent randomness (which makes it difficult to write deterministic tests), long computation times, and the selection of parameters governing the MCMC iteration itself. These issues are discussed in this section.

Here, we describe an MCMC framework based on Gibbs sampling for the problem of state estimation in spaSPT data. Building on the models introduced in the previous section, we introduce a Bayesian treatment of the estimation problem, which is important for regularizing the Gibbs sampling technique.

While EM or radial jump methods are generally more practical for finite-state mixture models, this discussion lays the groundwork for the more non-traditional MCMC methods necessary to implement the models in the next chapter.

4.3.1 Bayesian framework for finite-state diffusive mixtures

As before, suppose we have a set of N trajectories that we denote $\mathbb{X}$, so that the i^{th} trajectory is $\mathbf{X}_i \in \mathbb{X}$. We assume that each trajectory is associated with a state $Z_i \in \{1, ..., K\}$. Each state $j \in \{1, ..., K\}$ is characterized by a set of one or more

state parameters θ_j. The fractional occupancy of the j^{th} state across the whole dataset is τ_j.

Our goal is to evaluate the probability of the model parameters $\boldsymbol{\tau}$ and $\boldsymbol{\theta}$ given the observed trajectories using Bayes' theorem:

$$\Pr\left(\boldsymbol{\tau}, \boldsymbol{\theta} \mid \mathbb{X}\right) = \frac{\mathcal{L}\left[\boldsymbol{\tau}, \boldsymbol{\theta} \mid \mathbb{X}\right] \Pr\left(\boldsymbol{\tau}, \boldsymbol{\theta}\right)}{\Pr\left(\mathbb{X}\right)}$$

Here, $\Pr\left(\boldsymbol{\tau}, \boldsymbol{\theta}\right)$ is the *prior probability* for the model parameters, $\Pr\left(\mathbb{X}\right)$ is a normalization factor known as the *evidence*, and the likelihood $\mathcal{L}\left[\boldsymbol{\tau}, \boldsymbol{\theta} \mid \mathbb{X}\right]$ was introduced in the previous section (equation 4.6). The left-hand side is the *posterior probability* for the mixture parameters $\boldsymbol{\tau}$ and $\boldsymbol{\theta}$. Because the prior and the posterior are usually probability densities rather than discrete masses, we'll write this as

$$\pi\left(\boldsymbol{\tau}, \boldsymbol{\theta} \mid \mathbb{X}\right) = \frac{\mathcal{L}\left[\boldsymbol{\tau}, \boldsymbol{\theta} \mid \mathbb{X}\right] \pi\left(\boldsymbol{\tau}, \boldsymbol{\theta}\right)}{\Pr\left(\mathbb{X}\right)} \tag{4.18}$$

where the prior $\pi\left(\boldsymbol{\tau}, \boldsymbol{\theta}\right)$ and the posterior $\pi\left(\boldsymbol{\tau}, \boldsymbol{\theta} \mid \mathbb{X}\right)$ are treated as probability densities. This scheme is illustrated as a graphical model in Fig. 4.11.

In order to define the prior $\pi\left(\boldsymbol{\tau}, \boldsymbol{\theta}\right)$, we assume that $\boldsymbol{\tau}$ and $\boldsymbol{\theta}$ are drawn from some factorizable distribution:

$$\pi\left(\boldsymbol{\tau}, \boldsymbol{\theta} \mid \boldsymbol{\alpha}, H\right) = \pi\left(\boldsymbol{\tau} \mid \boldsymbol{\alpha}\right) \pi\left(\boldsymbol{\theta} \mid H\right)$$

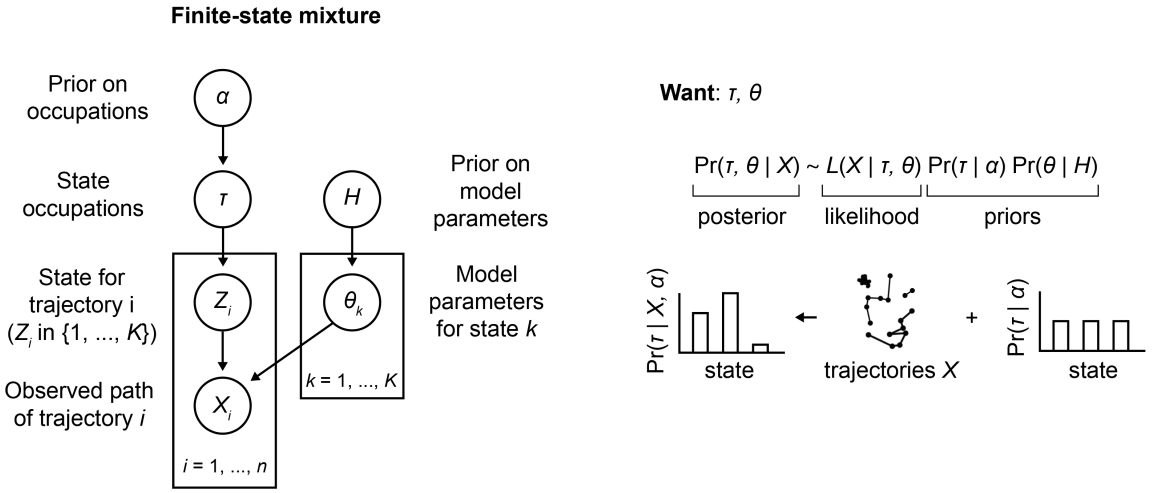

Figure 4.11: Graphical model for a Bayesian model for a finite-state mixture of diffusive states. Each state *j* is associated with a fractional occupancy τ_j and a set of one or more state parameters θ_j. Boxes represent *plates*, which represent sets of random variables with the same conditional structure. The goal of inference is to estimate the vectors $\boldsymbol{\tau}$ and $\boldsymbol{\theta}$.

$\boldsymbol{\alpha}$ and H are hyperparameters for the priors. For the prior over $\boldsymbol{\tau}$, we choose

$$\boldsymbol{\tau} \mid \boldsymbol{\alpha} \sim \text{Dirichlet}\,(\boldsymbol{\alpha})$$

The Dirichlet prior is a natural choice for discrete probabilities like $\boldsymbol{\tau}$. Because it is the conjugate prior for the multinomial distribution, it simplifies sampling for MCMC methods. For instance, if we have a multinomial random vector $\mathbf{n} \sim \text{Mult}\,(\boldsymbol{\tau}, N)$ where $\boldsymbol{\tau} \sim \text{Dirichlet}\,(\boldsymbol{\alpha})$, then Bayes' theorem gives us

$$\boldsymbol{\tau} \mid \mathbf{n} \sim \text{Dirichlet}\,(\boldsymbol{\alpha} + \mathbf{n})$$

This makes it clear that the hyperparameter $\boldsymbol{\alpha}$ acts as a set of *pseudocounts* for each state. As we accumulate more data (that is, as N increases), the magnitude of $\mathbf{n}$ will become much larger than the magnitude of $\boldsymbol{\alpha}$ and the prior will contribute less weight to the posterior estimate of $\boldsymbol{\tau}$.

Motivated by this interpretation, we'll usually choose to represent the hyperparameter in the form

$$\boldsymbol{\alpha} = \left(\frac{\alpha}{K}, ..., \frac{\alpha}{K}\right)^T \tag{4.19}$$

Since all of the elements of $\boldsymbol{\alpha}$ are equal, this imposes no favoritism onto the states. α represents the total pseudocounts in the prior. An important quantity is the *pseudocount fraction* $\alpha/(\alpha + N)$, which determines the relative strength of the prior against the data.

The prior over $\boldsymbol{\theta}$ will depend on the exact nature of the diffusion model. Consider the specific case of regular Brownian motion as an example. For RBM, θ_j is the diffusion coefficient for state j. Because the likelihood for RBM can be represented as a product of gamma distributions (eq. 3.17), the conjugate prior over θ_j would be another gamma distribution. This certainly simplifies the math, but using a gamma prior will lead to sampling some diffusion coefficients far more often than others. This isn't a very good reflection of our prior beliefs; we usually have no idea what $\boldsymbol{\theta}$ is in advance. An alternative choice is

$$\theta_j \sim \text{Uniform}\,(\theta_{\text{min}}, \theta_{\text{max}}) \tag{4.20}$$

In real data, $\theta_{\text{min}} = 0$ and $\theta_{\text{max}} = 100\ \mu\text{m}^2\ \text{s}^{-1}$, which bracket the range of protein diffusion coefficients observed in live cells to date, are good choices. Then the prior becomes

$$\pi\left(\theta_j\right) = \begin{cases} 1/(\theta_{\text{max}} - \theta_{\text{min}}) & \text{if } \theta_j \in [\theta_{\text{min}}, \theta_{\text{max}}] \\ 0 & \text{otherwise} \end{cases}$$

for all states j.

4.3.2 Rationale for the Gibbs sampler

As in the case of EM, the incomplete likelihood in eq. 4.18 poses challenges for inference. Inherent in the likelihood 4.6 is a marginalization over $\mathbf{Z}$. There are N elements in $\mathbf{Z}$, each of which can assume one of K different values, for a total of K^N terms. This quickly becomes unfeasible. We usually have thousands of trajectories or more ($N > 1000$) in each dataset.

Knowledge of $\mathbf{Z}$ would dramatically simplify the problem:

$$\pi\left(\boldsymbol{\tau}, \boldsymbol{\theta} \mid \mathbf{Z}, \mathbb{X}\right) = \frac{\mathcal{L}\left[\boldsymbol{\tau}, \boldsymbol{\theta} \mid \mathbf{Z}, \mathbb{X}\right] \pi\left(\boldsymbol{\tau} \mid \boldsymbol{\alpha}\right) \pi\left(\boldsymbol{\theta} \mid H\right)}{\Pr\left(\mathbf{Z}, \mathbb{X}\right)} \tag{4.21}$$

The likelihood function in eq. 4.21 corresponds to the "complete" likelihood in eq. 4.7, which is far more tractable than the incomplete likelihood 4.6.

The problem, of course, is that we don't actually know the state assignments $\mathbf{Z}$. So let's pull the same trick, but now in reverse: suppose that we know $\boldsymbol{\tau}$ and $\boldsymbol{\theta}$, but not $\mathbf{Z}$. Then Bayes' theorem becomes

$$\pi\left(\mathbf{Z} \mid \boldsymbol{\tau}, \boldsymbol{\theta}, \mathbb{X}\right) = \frac{\mathcal{L}\left[\mathbf{Z} \mid \boldsymbol{\tau}, \boldsymbol{\theta}, \mathbb{X}\right] \pi\left(\mathbf{Z}\right)}{\Pr\left(\mathbb{X}\right)} \tag{4.22}$$

where

$$\mathcal{L}\left[\mathbf{Z} \mid \boldsymbol{\tau}, \boldsymbol{\theta}, \mathbb{X}\right] = \prod_{i=1}^{N} \tau_{Z_i} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j\right)$$

$f_{\mathbf{X}|Z}(\mathbf{x} \mid j, \theta_j)$ is the probability density for a single trajectory in state j. We will usually take a noninformative prior for $\pi\left(\mathbf{Z}\right)$.

Unlike the joint posterior $\pi\left(\boldsymbol{\tau}, \boldsymbol{\theta}, \mathbf{Z} \mid \mathbb{X}\right)$, the *conditional* posteriors 4.21 and 4.22 are highly amenable to sampling. This suggests the following scheme:

1. Guess some initial $\boldsymbol{\tau}^{(0)}$ and $\boldsymbol{\theta}^{(0)}$.
2. For each iteration $t = 1, 2, ...$:
   (a) Conditioning on $\boldsymbol{\tau}^{(t-1)}$ and $\boldsymbol{\theta}^{(t-1)}$, sample $\mathbf{Z}^{(t)}$ from eq. 4.22.
   (b) Conditioning on $\mathbf{Z}^{(t)}$, sample $\boldsymbol{\tau}^{(t)}$ and $\boldsymbol{\theta}^{(t)}$ from eq. 4.21.

This scheme is the central idea of the MCMC method known as *blocked Gibbs sampling*. Proofs for its correctness generally need to demonstrate two things: first, that sequentially sampling from conditional distributions actually produces samples from the joint posterior, and second, that the Markov chain defined by the samples $\boldsymbol{\tau}^{(t)}$, $\boldsymbol{\theta}^{(t)}$, $\mathbf{Z}^{(t)}$ is ergodic on the joint posterior density. These proofs

has been discussed in great detail elsewhere - in particular, we like Casella and George's 1992 review [76] and Bishop's book [48], which both have plenty of examples.

One aspect of these proofs is worth highlighting here. A sufficient criterion for the MC to be ergodic is that neither of the conditional distributions 4.21 and 4.22 are anywhere zero. This means that any point in the parameter space is accessible from any other point. All of the diffusion models we have discussed so far satisfy this requirement. However, just because another point in the parameter space is *accessible* does not mean it is *easy to get to*.

To highlight this problem, consider the issue of sampling $\theta_j^{(t)}$, given $\mathbf{Z}^{(t)}$ and $\mathbb{X}$. Since $\mathbf{Z}^{(t)}$ gives us the exact set of trajectories belonging to state j, the component of 4.21 corresponding to θ_j is

$$\pi\left(\theta_j \mid \mathbf{Z}, \mathbb{X}\right) \propto \pi\left(\theta_j\right) \prod_{i=1}^{N} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j\right)^{\mathbb{I}_{Z_i=j}}$$

where $\pi(\theta_j)$ is the prior from 4.20. Since this prior is not conjugate to the likelihood $f_{\mathbf{X}|Z}(\mathbf{x})$, we have no direct analytical representation for the posterior $\pi(\theta_j|\mathbf{Z}, \mathbb{X})$. Instead, we must sample from the posterior numerically, most commonly by a Metropolis-Hastings step:

1. Generate θ_j^* according to some proposal distribution $g(\theta_j^* \mid \theta_j^{(t)})$. The most common choice for the proposal distribution is

$$\theta_j^* \mid \theta_j^{(t)} \sim \mathcal{N}\left(\theta_j^{(t)}, \nu^2\right)$$

2. Evaluate the acceptance ratio

$$\begin{aligned} r &= \frac{g(\theta_j^*|\theta_j^{(t)})}{g(\theta_j^{(t)}|\theta_j^*)} \frac{\pi\left(\theta_j^* \mid \mathbf{Z}, \mathbb{X}\right)}{\pi\left(\theta_j^{(t)} \mid \mathbf{Z}, \mathbb{X}\right)} \\ &= \frac{g(\theta_j^{(t)}|\theta_j^*)}{g(\theta_j^*|\theta_j^{(t)})} \frac{\prod_{i=1}^{N} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j^*\right)^{\mathbb{I}_{Z_i=j}}}{\prod_{i=1}^{N} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j^{(t)}\right)^{\mathbb{I}_{Z_i=j}}} \end{aligned}$$

3. Sample $u \sim \text{Uniform}(0, 1)$. If $u < r$, set $\theta_j^{(t+1)} = \theta_j^*$. Otherwise, set $\theta_j^{(t+1)} = \theta_j^{(t)}$.

Several points are worth noting about this scheme. First, notice how the noninformative prior 4.20 cancels, effectively exerting no influence on the result. Second, when we use Gaussian-distributed steps like this, the bias term becomes

$$\frac{g(\theta_j^{(t)}|\theta_j^*)}{g(\theta_j^*|\theta_j^{(t)})} = \frac{\Phi\left(\frac{\theta_{\max}-\theta_j^{(t)}}{\nu}\right) - \Phi\left(\frac{\theta_{\min}-\theta_j^{(t)}}{\nu}\right)}{\Phi\left(\frac{\theta_{\max}-\theta_j^*}{\nu}\right) - \Phi\left(\frac{\theta_{\min}-\theta_j^*}{\nu}\right)} \tag{4.23}$$

where $\Phi(x)$ is the CDF for a Gaussian with zero mean and unit variance.

The most crucial part of this scheme is the selection of ν^2, the variance of the steps. While any choice of ν^2 leads to a correctly ergodic Markov chain, some ν^2 lead to more efficient samplers than others. If the steps are too large, then few of them will land on values of θ_j that are feasible given the current $\mathbf{Z}$. As a result, the proposal θ_j^* will only rarely be accepted and exploration of the parameter space will be inefficient. On the other hand, if ν^2 is too small, then the Markov chain will take a high number of iterations to traverse any significant distance in parameter space; the samples will have a high autocorrelation.

This demonstrates the point mentioned above: while points in parameter space may be *accessible*, they may not be *easy to get to*. We can overcome such issues by using more iterations, but for very large datasets this may be unfeasible. This is a fundamental limitation of the Gibbs sampling approach.

4.3.3 Gibbs sampler for regular Brownian motion

As an example, we apply the Gibbs sampler to regular Brownian motion (RBM) observed on a microcope with a shallow depth of field Δz. If each trajectory i has L_i jumps in m spatial dimensions and the sum of squared radial jumps is S_i, then the trajectory likelihood is (using eq. 3.17)

$$f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j\right) = \frac{S_i^{\frac{mL_i}{2}-1} \exp\left(-\frac{S_i}{4(\theta_j \Delta t + \sigma_{\text{loc}}^2)}\right)}{\Gamma\left(\frac{mL_i}{2}\right)\left(4(\theta_j \Delta t + \sigma_{\text{loc}}^2)\right)^{\frac{mL_i}{2}}}$$

As before, we assume that defocalization imposes some bias on the observed fractional occupancies. If η_j is the probability to observe a jump from a trajectory in state j, then the *observed* occupancy of state j is $\mu_j \propto \eta_j \tau_j$.

Algorithm 4.4 is a straightforward implementation of this scheme. Notice that the output of the method is a sequence of samples from the posterior distribution.

Algorithm 4.4: Gibbs sampling for a finite-state mixture of regular Brownian motions

Parameters: $\mathbb{X}$, a set of N experimentally observed trajectories; K, the number of diffusive states; α, the number of pseudocounts; ν^2, the Metropolis step variance; θ_{min} and θ_{max}, bounds on the acceptable diffusion coefficient; σ^2_{loc}, the localization error in μm^2; Δt, the frame interval; Δz, the focal depth.

Algorithm:

1. Generate some initial guesses $\boldsymbol{\tau}^{(0)} \sim \text{Dirichlet}\left(\frac{\alpha}{K}, ..., \frac{\alpha}{K}\right)$ and $\theta_j \sim \text{Uniform}\left(\theta_{\text{min}}, \theta_{\text{max}}\right)$ for $j = 1, ..., K$.
2. For each iteration $t = 1, 2, ...$:
   (a) For each trajectory $i = 1, ..., N$, draw $Z_i \sim \text{Mult}\left(\mathbf{p}_i, N\right)$ where
   $$p_{i,j} \propto \frac{\tau_j^{(t-1)}}{\eta_j} f_{\mathbf{X}|Z}(\mathbf{X}_i | Z_i = j, \theta_j^{(t-1)})$$
   (b) Draw a new state occupation vector:
   $$\boldsymbol{\tau}^{(t)} \mid \mathbf{Z}^{(t)} \sim \text{Dirichlet}\left(n_1 + \frac{\alpha}{K}, ..., n_K + \frac{\alpha}{K}\right)$$
   where $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_i^{(t)}=j}$.
   (c) For each state $j = 1, ..., K$:
      i. Propose a new diffusion coefficient $\theta_j^* \sim \mathcal{N}\left(\theta_j^{(t-1)}, \nu^2\right)$.
      ii. Calculate the acceptance ratio
      $$r = \frac{\Phi\left(\frac{\theta_{\text{max}} - \theta_j^{(t)}}{\nu}\right) - \Phi\left(\frac{\theta_{\text{min}} - \theta_j^{(t)}}{\nu}\right)}{\Phi\left(\frac{\theta_{\text{max}} - \theta_j^*}{\nu}\right) - \Phi\left(\frac{\theta_{\text{min}} - \theta_j^*}{\nu}\right)} \frac{\prod_{i=1}^{N} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j^*\right)^{\mathbb{I}_{Z_i=j}}}{\prod_{i=1}^{N} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j^{(t)}\right)^{\mathbb{I}_{Z_i=j}}}$$
      iii. Draw $u \sim \text{Uniform}(0, 1)$. If $u < r$, set $\theta_j^{(t)} = \theta_j^*$; otherwise set $\theta_j^{(t)} = \theta_j^{(t-1)}$.
3. Return the sequence of samples $\left(\boldsymbol{\tau}^{(t)}, \boldsymbol{\theta}^{(t)}, \mathbf{Z}^{(t)}\right)_{t \in \mathbb{N}}$.

Gibbs sampler for fractional Brownian motion

Algorithm 4.4 is easily tweaked to accommodate fractional Brownian motion rather than regular Brownian motion. The ability of Gibbs sampling to generalize easily to non-Brownian models is a major advantage over methods like EM.

The likelihood function for FBM cannot be boiled down to a single statistic analogous to the sum of squared jumps for RBM. As a result, we'll need to make some notational changes to accommodate it.

Represent each trajectory $\mathbf{X}_i \in \mathbb{X}$ as the set of 1D jumps along each spatial dimension, so that $X_{i,d,j}$ represents the j^{th} jump in the d^{th} spatial dimension for the i^{th} trajectory. Let L_i be the number of jumps in trajectory i.

For the state parameters, let $\theta_j = (D_j, H_j)$ where D_j and H_j are the diffusion coefficient and Hurst parameter for state j, respectively. Then replace $f_{\mathbf{X}|Z}(\mathbf{x})$ with the density function generated by eq. 3.31, which in m spatial dimensions is

$$f_{\mathbf{X}|Z}(\mathbf{X}_i \mid Z_i = j, \theta_j) = \frac{\exp\left(-\frac{1}{2}\sum_{d=1}^{m} \mathbf{X}_{i,d}^T C_\Delta^{-1} \mathbf{X}_{i,d}\right)}{(2\pi)^{mL_i/2} \det(C_\Delta)^{\frac{m}{2}}}$$

with

$$\begin{aligned}(C_\Delta)_{kl} = \overline{D}_j \Delta t \left(|k-l+1|^{2H_j} + |k-l-1|^{2H_j} - 2|k-l|^{2H_j}\right) \\ + 2\sigma_{\text{loc}}^2 \mathbb{I}_{i=j} - \sigma_{\text{loc}}^2 \mathbb{I}_{|i-j|=1}\end{aligned}$$

and $\overline{D}_j = D_j \Delta t^{2H_j - 1}$ is the modified diffusion coefficient, discussed in Appendix B.

The Metropolis-Hastings step (step 2(c) in Algorithm 4.4) can then be replaced with the joint proposal

$$\begin{aligned}H_j^* &\sim \mathcal{N}\left(H_j^{(t-1)}, \nu_H^2\right) \\ D_j^* &\sim \mathcal{N}\left(D_j^{(t-1)}, \nu_D^2\right)\end{aligned}$$

Then acceptance ratio then becomes

$$r = \left(\frac{\Phi\left(\frac{D_{\max} - D_j^{(t)}}{\nu_D}\right) - \Phi\left(\frac{D_{\min} - D_j^{(t)}}{\nu_D}\right)}{\Phi\left(\frac{D_{\max} - D_j^{*}}{\nu_D}\right) - \Phi\left(\frac{D_{\min} - D_j^{*}}{\nu_D}\right)} \right) \left(\frac{\Phi\left(\frac{1 - H_j^{(t)}}{\nu_H}\right) - \Phi\left(\frac{-H_j^{(t)}}{\nu_H}\right)}{\Phi\left(\frac{1 - H_j^{*}}{\nu_H}\right) - \Phi\left(\frac{-H_j^{*}}{\nu_H}\right)} \right) \cdot \left(\frac{\prod_{i=1}^{N} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j^{*}\right)^{\mathbb{I}_{Z_i=j}}}{\prod_{i=1}^{N} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid Z_i = j, \theta_j^{(t)}\right)^{\mathbb{I}_{Z_i=j}}} \right)$$

The rest of the algorithm proceeds as before. Importantly, if we hold $H = 0.5$, then this algorithm also provides a means to estimate the localization error, which can be defined on a state-by-state basis.

4.3.4 Posterior point estimates

The output of Gibbs sampling as represented in Algorithm 4.4 is a set of samples of the joint posterior distribution $\left(\boldsymbol{\tau}^{(t)}, \boldsymbol{\theta}^{(t)}, \mathbf{Z}^{(t)}\right)_{t \in \mathbb{N}}$. Typically we want to obtain some kind of point estimates $\hat{\boldsymbol{\tau}}$ and $\hat{\boldsymbol{\theta}}$ from these samples.

Two options are most common: the *maximum a posteriori* estimate and the *posterior mean*.

Maximum a posteriori estimate

The maximum a posteriori (MAP) estimate is defined as the point in the marginal posterior distribution with the maximum posterior probability. For $\boldsymbol{\tau}$, this definition is

$$\hat{\boldsymbol{\tau}}_{\text{MAP}} = \underset{\boldsymbol{\tau}}{\text{argmax}}\, \pi\left(\boldsymbol{\tau} \mid \mathbb{X}\right)$$

and likewise for $\boldsymbol{\theta}$. Since we only have discrete samples from the posterior distribution, it is necessary either to bin the posterior distribution or use some kind of kernel density estimate. The parameters governing binning and the KDE are choices that the experimentalist must make.

Posterior mean

In contrast to the MAP estimate, the posterior mean is nonparametric:

$$\hat{\boldsymbol{\tau}}_{\text{mean}} = \mathbb{E}\left[\boldsymbol{\tau}\right] = \frac{1}{\text{\# iterations}} \sum_{\text{iter } t} \boldsymbol{\tau}^{(t)}$$

and likewise for θ.

Because it is simpler and tends to provide more conservative estimates of model parameters, we use the posterior mean in this thesis. Typically, we only include samples after a certain "burn-in" period. Gibbs samplers of the type introduced in Algorithm 4.4 frequently converge rapidly to the posterior distribution, so usually 20 - 100 iterations are sufficient for the burn-in period.

4.3.5 Identifiability

The most serious issue with the Gibbs sampling approach is that the states are not unique. To see this, imagine that we have a two-component ($K = 2$) regular Brownian motion model, and that there is a maximum in the posterior probability at

$$\begin{aligned} \tau_1 &= 0.3 \\ \tau_2 &= 0.7 \\ D_1 &= 0.1\ \mu\text{m}^2\ \text{s}^{-1} \\ D_2 &= 1.0\ \mu\text{m}^2\ \text{s}^{-1} \end{aligned}$$

Then there is necessarily another maximum at

$$\begin{aligned} \tau_1 &= 0.7 \\ \tau_2 &= 0.3 \\ D_1 &= 1.0\ \mu\text{m}^2\ \text{s}^{-1} \\ D_2 &= 0.1\ \mu\text{m}^2\ \text{s}^{-1} \end{aligned}$$

The two models differ only in the labels assigned to the two states, which have no intrinsic meaning apart from notational convenience. The model is invariant under any permutation of these labels. Since there are $K!$ permutations for K labels, a mixture of K diffusing states will have at least $K!$ maxima in the posterior distribution that are equivalent. This is known as the *identifiability* or *label-switching* problem, and has been reviewed in the context of mixture models in [77].

There are numerous ways to deal with identifiability, some more elegant than others. The simplest solution is to impose a constraint on the diffusion coefficients - for instance, requiring that $D_2 \geq D_1$ in the model above. However, this seemingly innocuous assumption can have severe consequences for inference and can even contradict the prior on D.

An alternative approach is to compute the posterior mean, and then reorder the components in terms of some condition - for instance, strictly increasing diffusion

coefficient. This does not work when the Gibbs sampler traverses multiple modes during the course of inference, but this is rarely the case for real spaSPT data and so it mostly suffices for our purposes.

The approaches outlined in the next chapter remove most of the identifiability concerns raised here, which can be considered an artifact of considering K discrete states.

4.4 Radial jump histogram-based estimators

The EM algorithm and Gibbs sampler considered previously are two ways to analyze mixtures of diffusive states. Another is to fit the empirical jump length histogram. (Usually, fitting to the empirical distribution function/CDF is more appropriate, since it limits discretization artifacts associated with binning.) This approach has been discussed in detail for the specific case of regular Brownian motion by Mazza [59] and Hansen & Woringer [60], so here we only provide a high-altitude overview.

Suppose, as before, that we have a mixture of K diffusing states characterized by the state occupation vector $\boldsymbol{\tau} = (\tau_1, ..., \tau_K)^T$ and the diffusive parameters $\boldsymbol{\theta} = (\theta_1, ..., \theta_K)^T$. Let $\mathbf{R} \in \mathbb{R}^m$ be a random jump made by a particle taken from this mixture over n successive frame intervals, and let $R = |\mathbf{R}|$ be the end-to-end radial distance of this jump.

If we know that the jump is made from some diffusive state j with parameters θ_j, then we can write the CDF for R as

$$F_R\left(r, n \mid \text{state } j, \theta_j, \Delta t, \sigma^2_{\text{loc}}, m\right) = \Pr\left(R \leq r \mid \theta_j, n\Delta t, \sigma^2_{\text{loc}}, m\right)$$

For example, if we have regular Brownian motion in two dimensions ($m = 2$) after three frames ($n = 3$), we can apply eq. 3.10 to get the CDF

$$F_R\left(r, n = 3\right) = 1 - \exp\left(-\frac{r^2}{4(n\theta_j\Delta t + \sigma^2_{\text{loc}})}\right)$$

If we don't know the state that R comes from, the CDF for a jump taken from the mixture is

$$F_R\left(r, n \mid \boldsymbol{\tau}, \boldsymbol{\theta}, \Delta t, \sigma^2_{\text{loc}}, m\right) = \sum_{j=1}^{K} \tau_j F_R\left(r \mid \text{state } j, \theta_j, n\Delta t, \sigma^2_{\text{loc}}, m\right) \tag{4.24}$$

This equation assumes that we sample in an unbiased manner from all of the states in the mixture, neglecting the effects of defocalization. It is straightforward to

incorporate defocalization corrections. Suppose as before that $\eta_j(n)$ is the probability that our microscope observes a jump from a trajectory in state j after n frame intervals. $\eta_j(n)$ is a function of the diffusive parameters θ_j, the frame interval Δt, and the focal depth Δz, and can be computed with Algorithms 4.1 or 4.2, as appropriate for the diffusion model and the number of gaps. If we let

$$\mu_j = \frac{\eta_j \tau_j}{\sum_{k=1}^{K} \eta_k \tau_k}$$

be the *apparent* fraction of the j^{th} state in the mixture after the effects of defocalization, then we have the mixture CDF

$$F_R\left(r, n \mid \boldsymbol{\tau}, \boldsymbol{\theta}, \Delta t, \Delta z, \sigma^2_{\text{loc}}, m\right) = \sum_{j=1}^{K} \mu_j(\theta_j, \Delta t, \Delta z) F_R\left(r \mid \text{state } j, \theta_j, n\Delta t, \sigma^2_{\text{loc}}, m\right) \tag{4.25}$$

As a simple example, consider a two-component mixture of regular Brownian motion states in two spatial dimensions ($m = 2$). Then, dropping the parameter dependences for clarity, we would have

$$F_R(r, n) = 1 - \frac{1}{\eta_1(n)\tau_1 + \eta_2(n)\tau_2}\left(\eta_1(n)\tau_1 e^{-\frac{r^2}{4(n\theta_1\Delta t + \sigma^2_{\text{loc}})}} + \eta_2(n)\tau_2 e^{-\frac{r^2}{4(n\theta_2\Delta t + \sigma^2_{\text{loc}})}}\right)$$

As another example, if we replace the two RBM states with two FBM states characterized by diffusion coefficients D_1 and D_2 and Hurst parameters H_1 and H_2, then we would have

$$F_R(r, n) = 1 - \frac{1}{\eta_1(n)\tau_1 + \eta_2(n)\tau_2}\left(\eta_1(n)\tau_1 e^{-\frac{r^2}{4(nD_1\Delta t^{H_1} + \sigma^2_{\text{loc}})}} + \eta_2(n)\tau_2 e^{-\frac{r^2}{4(nD_2\Delta t^{H_2} + \sigma^2_{\text{loc}})}}\right)$$

The function $F_R(r, n)$ is fit with respect to the parameters $\boldsymbol{\tau}$ and $\boldsymbol{\theta}$, and potentially σ^2_{loc} if desired.

The mathematical simplicity of CDF fitting makes it particularly suited to dealing with complex diffusion models without efficient maximum likelihood estimators, such as Levy flights. However, the approach has some serious limitations:

1. The CDF fitting approach is not guaranteed to converge to a likelihood maximum.

2. The approach is dependent on the jump binning scheme. If the jump bins have linear spacing, then longer jumps have a stronger influence on the result than shorter jumps due to the higher number of points in the tail of the CDF.

3. By discarding all connectivity information between subsequent jumps in a trajectory, the approach has limited inferential utility for complex mixtures of states or diffusion models with memory (such as FBM).

4. Because the CDF fitting approach is not phrased in terms of probability, it cannot assign component likelihoods to individual trajectories. In other words, we cannot go back to the original set of trajectories and estimate how likely each trajectory is to have to come from a particular diffusive state.

The last point is particularly relevant when we want to understand whether trajectories have different diffusive properties in different parts of the cell. Without being able to relate individual trajectories to the likelihoods of different states, we lose all of the spatial information inherent in raw spaSPT data.

4.5 Comparison of estimators for finite-state mixtures

We have described three frameworks for inference on finite-state mixtures of diffusing states - expectation maximization, Gibbs sampling, and radial jump histogram fitting. Are these approaches equivalent solutions to the problem?

We compared the number of trajectories required for convergence for the three different approaches (Fig. 4.12), finding that all three approaches converged at a similar number of trajectories to the true model parameters. After 1000 trajectories, little additional improvements on accuracy were made by any trajectory.

In contrast, on real data, the three methods show more variability. This variability is small but systematic when using a two-state model (Fig. 4.13), while it becomes more considerable when analyzing data with a three-state model (Fig. 4.14). Two- or three-state models are approximations for trajectories gathered from real cells, which reflect molecules in perhaps dozens or hundreds of distinct diffusive states. The deviations between models reflect the degree to which these methods handle this complexity.

The comparisons in Figs. 4.13 and 4.14 demonstrate an important challenge for SPT methods: how to deal with complex input. Some methods - in particular radial jump fitting methods - take advantage of the full complexity of the models that we fit with. This results in instability when fitting in higher-dimensional parameter spaces. In contrast, other methods have a natural penalty on the complexity of the

model. This latter class of methods, which are mostly descended from Bayesian techniques, are the subject of the next chapter and address many of the concerns with the finite-state estimators compared here.

4.6 Some model selection concerns

In the last three chapters, we considered three categories of motion - regular Brownian motion, fractional Brownian motion, and Levy flights. For all three approaches, given the correct choice of model, it is possible to extract model parameters for that model. However, this does not address the issue of model selection

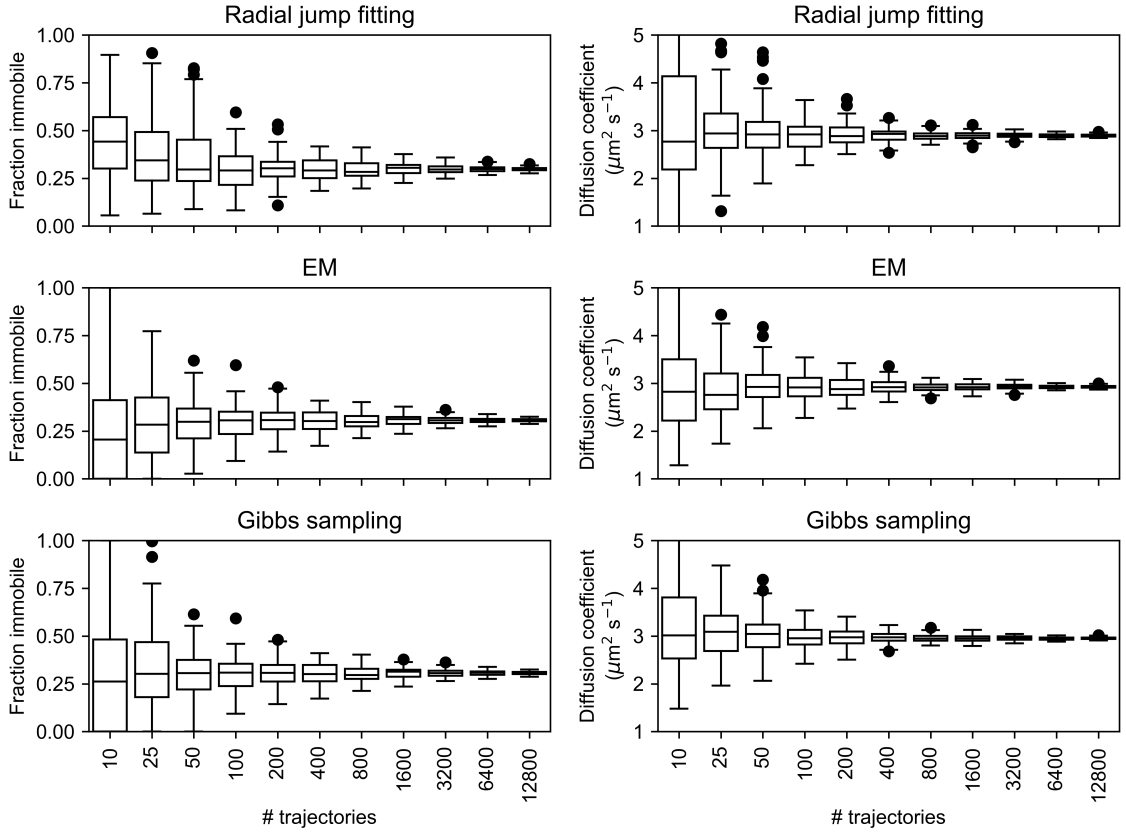

Figure 4.12: Comparison of the convergence efficiency for three different finite-state mixture estimators in simulated SPT. Trajectories were simulated in two diffusing states - a slow state (fraction 30%) with diffusion coefficient 0.01 μm^2 s^{-1} and a fast state (fraction 70%) with diffusion coefficient 3.0 μm^2 s^{-1}. Tracking was simulated in a thin focal volume (700 nm) bisecting a spherical nucleus with 5 μm radius, with 10 ms frame intervals, 30 nm localization error, and a 10 Hz bleaching rate. Box edges represent the 25th and 75th quantiles. 100 iterations were run per experiment.

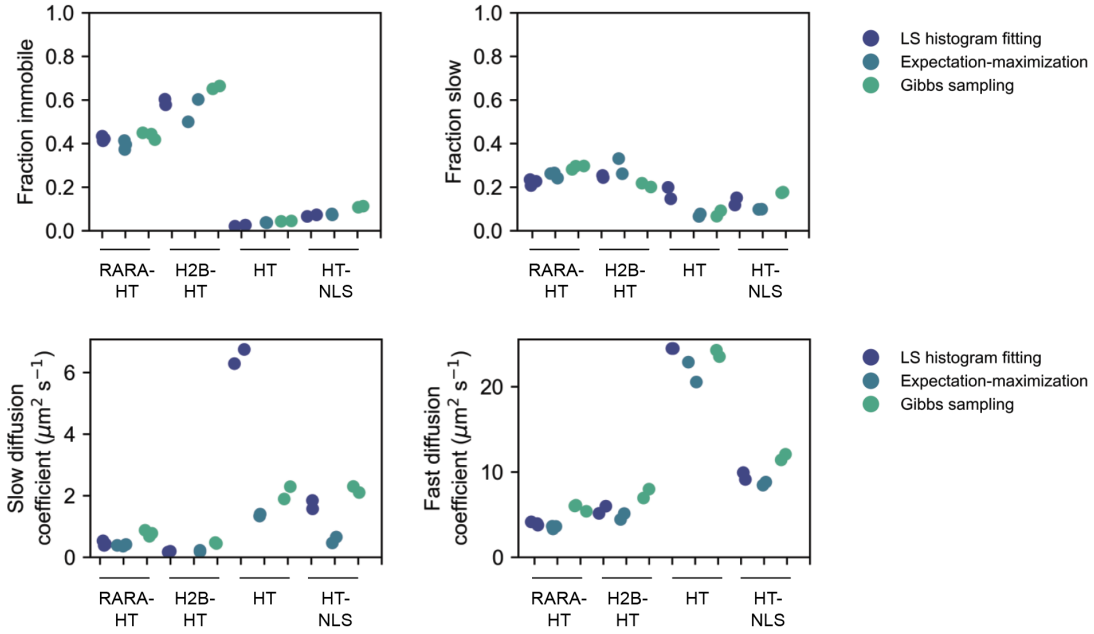

Figure 4.13: Comparison of three different finite-state mixture estimators on real trajectories with a two-state model. The constructs correspond to the following: *RARA-HT*, endogenously tagged retinoic acid receptor α-HaloTag-3xFLAG in U2OS nuclei; *H2B-HT*, stably transfected H2B-HaloTag-SNAPf in U2OS nuclei; *HT*, transiently transfected HaloTag-MCS in U2OS nuclei; *HT-NLS*, transiently transfected HaloTag-3xNLS in U2OS nuclei. Cells were labeled with 100 nM PA-JFX549 for 30 min, followed by four 30 min washes. Tracking was performed with 7.48 ms frame intervals, 1.5 ms pulse widths on microscope with approximately 700 nm depth of field and 160 nm pixels; the approximate 1D dimensional root positional variance associated with localization under these settings is $\sim$ 35 nm. Fits were performed without parameter constraints.

in the first place. Two questions are especially important:

1. How do we decide whether the mode of diffusion is Brownian or non-Brownian?
2. How do we decide how many diffusive states are present in our data?

The first point was addressed in section 3.2. Selecting the wrong model does have some consequences for the ability to infer state occupations - see Fig. 4.16. But the second point is equally important. Indeed, much of the biological interpretation of SPT revolves around assigning different diffusive states to one or another biochemical function. In the next chapter, we consider models that are geared toward resolving the second question.

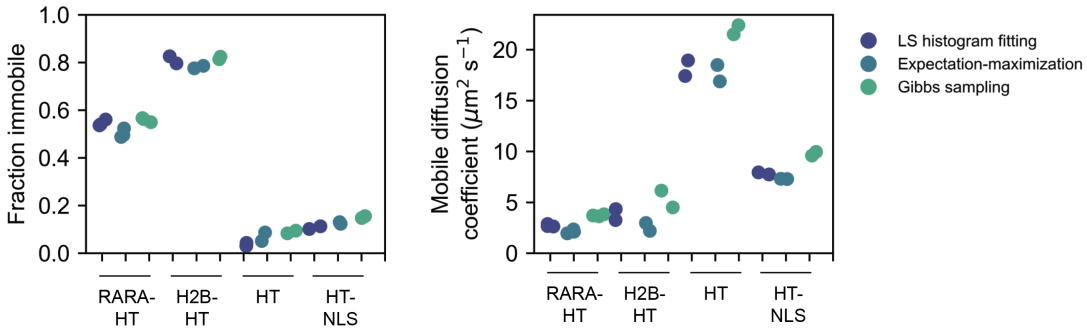

Figure 4.14: Comparison of three different finite-state mixture estimators on real trajectories with a three-state model. Sample preparation and tracking were performed as in Fig. 4.13. Fits were performed without parameter constraints. Individual dots indicate biological replicates.

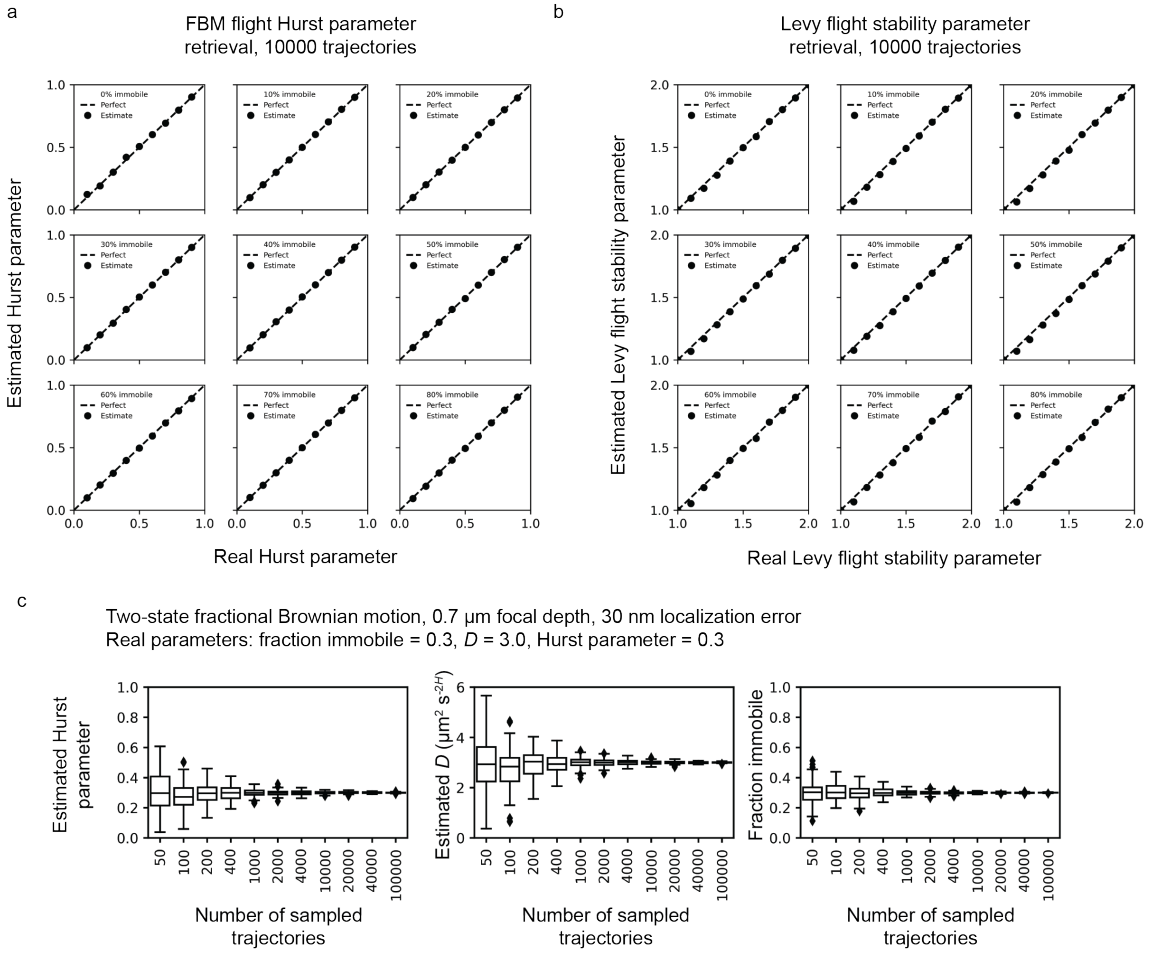

Figure 4.15: Using the radial jump histogram to extract the Hurst parameter from fractional Brownian motion and the stability parameter from Levy flights. Accuracy for parameter retrieval for non-Brownian diffusion models. (a) Estimating the Hurst parameter for sampled fractional Brownian motion (FBM) trajectories. 10000 trajectories were simulated in a nucleus observed with a synthetic focal depth of 0.7 μm in one of two diffusing states with diffusion coefficients different by two orders of magnitude, then fit at four frame intervals using the FBM diffusion model incorporating defocalization. (b) Estimating the stability parameter for simulated Levy flight trajectories 10000 trajectories were simulated in a nucleus observed with a synthetic focal depth of 0.7 μm in one of two diffusing states ("immobile" refers to the fraction of molecules in the slower-diffusing state), then fit at four frame intervals using the Levy flight diffusion model (c) Accuracy of parameter retrieval for two-state FBM at various numbers of sampled trajectories.

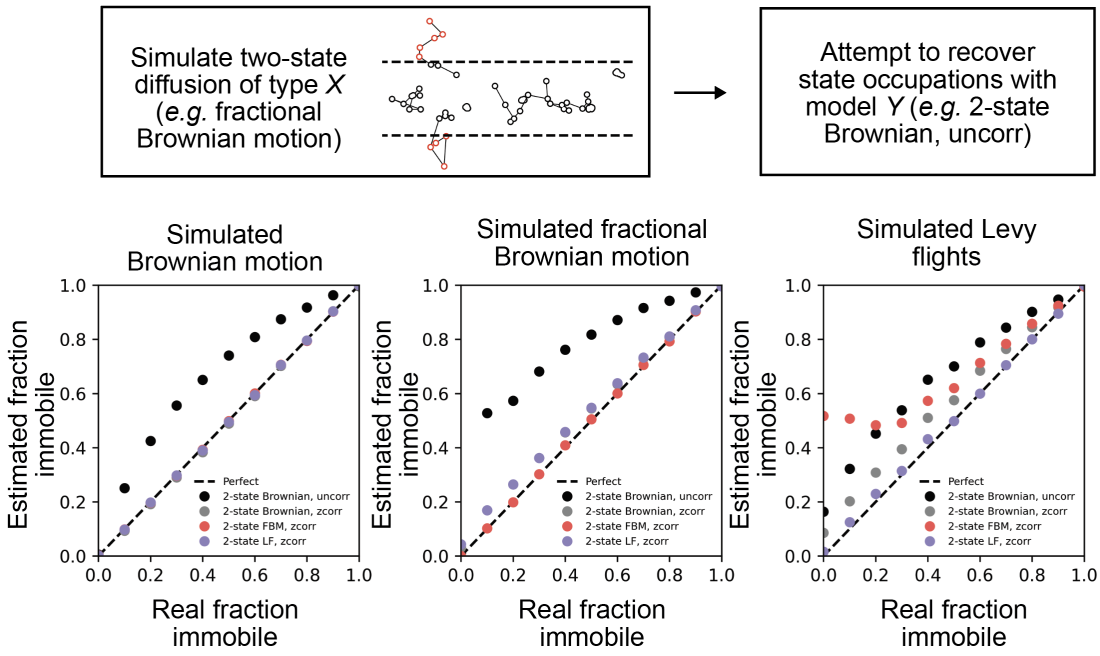

Figure 4.16: Assessing the accuracy of immobile fraction estimation when the model doesn't match the data. For each subplot, one of three categories of diffusion was simulated and then fit with a variety of other models. Estimates of the immobile fraction are generally less accurate when the real fraction immobile is low.

Chapter 5

Model selection

The previous two chapters presented some methods to infer the parameters governing mixtures of diffusing states. However, these chapters did not address the question of *how to select a particular mixture model from a set of alternatives*. In most of the cases dealt with so far, we assumed we *know* the underlying type of diffusion and the number of states, but this is rarely the case in experiments.

In this chapter, we consider methods to learn diffusion models from data. Given a particular set of trajectories, we seek not only to learn the occupations, diffusion coefficients, and anomaly parameters for the various states, but also to learn the *type of diffusion* and the *number of states* - a problem we refer to as *model selection*.

Some of the approaches previously outlined - for example, the jump histogram fitting method - tend to exploit all the available degrees of freedom in the model. This means that model selection cannot be informed by the data itself (unless we are willing to employ ad hoc methods like the Akaike or Bayes information criteria), but must depend on our prior beliefs about the states available to a molecule of interest. In many cases, however, we do not have access to this information. Indeed, learning how many distinct diffusive states a particular protein can inhabit in the cellular milieu is probably one of the most valuable pieces of information we could learn from spaSPT. The failure of jump length histogram methods to provide this information is one of their principal shortcomings.

The primary results of this chapter are a set of three algorithmic frameworks for model selection: discrete-state variational Bayes, arrayed state samplers, and Dirichlet processes. The latter two methods extend easily to non-Brownian diffusion models, but we also find that the discrete-state variational Bayes framework is a valuable drop-in replacement for jump length histogram methods.

5.1 Jump histogram-based methods

A few techniques for nonparametric recovery of the distribution of diffusion coefficients operate on the jump histogram, aggregated across all trajectories. Here, we highlight two of these as a counterpoint to the Bayesian methods considered later in the chapter.

5.1.1 Laplace transform methods

When trajectories are collected in two dimensions, a simple and naïve method to recover the distribution of the diffusion coefficient relies on the inverse Laplace transform of the jump length histogram.

Let $S = (X_{t+\Delta t} - X_t)^2 + (Y_{t+\Delta t} - Y_t)^2$ represent a single squared jump for a regular Brownian motion in two dimensions. We saw that the probability density for this jump conditional on some diffusion coefficient D is given by

$$\begin{aligned} S \mid D &\sim \text{Gamma}\left(1, \frac{1}{4(D\Delta t + \sigma_{\text{loc}}^2)}\right) \\ &= \text{Expon}\left(\frac{1}{4(D\Delta t + \sigma_{\text{loc}}^2)}\right) \end{aligned}$$

where σ_{loc}^2 is the localization error and Expon is the exponential density. Define the *spatial variance* $\phi = 4(D\Delta t + \sigma_{\text{loc}}^2)$, which collects the contributions to the observed jumps from both diffusion and localization error.

Then, if the true underlying distribution of ϕ is $f_\phi(\phi)$, the observed distribution of jumps is

$$\begin{aligned} f_S(s) &= \int_0^\infty f_{S|\phi}(s|\phi) f_\phi(\phi) \mathrm{d}\phi \\ &= \int_0^\infty \phi^{-1} f_\phi(\phi) e^{-s\phi^{-1}} \mathrm{d}\phi \end{aligned}$$

Let $p = \phi^{-1}$. Then this becomes

$$\begin{aligned} f_S(s) &= \int_0^\infty \left(\frac{f_\phi(p^{-1})}{p}\right) e^{-sp} \mathrm{d}p \\ &= \mathcal{L}\left[\frac{f_\phi(p^{-1})}{p}\right](s) \end{aligned}$$

where $\mathcal{L}$ is the Laplace transform. Recognizing that division by p in the real domain corresponds to integration in the Laplace domain, and accounting for the term at $p = 0$, we have

$$f_\phi(p^{-1}) = \mathcal{L}^{-1}\left[1 - F_S(s)\right](p) \tag{5.1}$$

Here, $F_S(s) = \int_0^s f_S(s')\mathrm{d}s'$ is the cumulative distribution function of S.

So in principle, we can get the distribution of diffusion coefficients by taking the inverse Laplace transform of the squared jump distribution. This is analogous to the use of the inverse Laplace transform to get the residence times for single molecules binding to chromatin, proposed recently [78]. Unfortunately, this approach is highly impractical. The inverse Laplace transform is numerically unstable and requires a great deal of regularization, especially since our actual distribution of S is discrete and usually has substantial noise associated with it. Indeed, the behavior at very low ϕ (very high p) can be entirely determined by a small number of bins at the lower end of the jump distribution. Gebhardt and coworkers have investigated these problems in detail for the residence time estimation problem [78], and most of their work is devoted to imposing regularization conditions on the ILT.

At a more fundamental level, using the squared jump distribution disregards most of the information inherent in trajectories. That is, we can learn more about a trajectory by considering it in its entirety than by decomposing it into a sequence of jumps.

5.1.2 Richardson-Lucy algorithm

A method proposed by Wang *et al.* [79] takes an alternative approach that is mathematically similar to but far more stable than the naïve inverse Laplace transform. This method is based on the Richardson-Lucy algorithm, which is more commonly used for image deconvolution.

As above, let ϕ be the spatial variance, collecting time-dependent and -independent contributions to the apparent particle position. Suppose that $f_{S,\mathrm{obs}}(s)$ is the experimentally observed jump histogram, that $f_\phi^{(t)}(\phi)$ is the estimated distribution of spatial variances at the t^{th} iteration, and that $f_{S,\mathrm{model}}^{(t)}(s)$ is the predicted jump histogram based on $f_\phi^{(t)}(\phi)$. That is,

$$f_{S,\mathrm{model}}(s) = \int_0^{\phi_{\max}} f_\phi^{(t)}(\phi) f_{S|\phi}(s|\phi)\mathrm{d}\phi$$

Then we can directly apply Lucy's method [80] to generate the iterative scheme

$$f_{\phi}^{(t+1)}(\phi) = f_{\phi}^{(t)}(\phi) \int \frac{f_{S,\text{obs}}(s)}{f_{S,\text{model}}^{(t)}(s)} f_{S|\phi}(s|\phi) \mathrm{d}S$$

In the light of 5.1, this method can be viewed as a way to regularize the inverse Laplace transform. As such, it requires a very large number of jumps to avoid sensitivity to noise and in practice takes a large number of iterations to converge. Like the inverse Laplace transform approach, this also disregards most of the information inherent in the trajectories by reducing them to a single jump histogram.

An alternative approach is to use Bayesian inference. As we will see, this also regularizes the inverse problem, but in a more principled way than for the inverse Laplace transform.

5.2 Discrete-state variational Bayes

The expectation-maximization (EM) algorithm described in the previous chapter relied on iterative maximization of the "merit function" (equation 4.10)

$$\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\tau}^{(t+1)} = \underset{\boldsymbol{\theta},\boldsymbol{\tau}}{\text{argmax}}\, \mathbb{E}\left[\log \mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\tau} | \mathbf{Z}, \mathbb{X}\right)\right]$$

In this criterion, $\mathbb{X}$ was an experimentally observed set of trajectories which could inhabit one of K different diffusive states with occupations $\boldsymbol{\tau}$ and state parameters (*e.g.* diffusion coefficients) $\boldsymbol{\theta}$.

Equation 4.10 was presented without justification. Here, we provide motivation for this scheme from the context of variational Bayesian (VB) statistics. Interpretation of EM in this context is well-established; chapter 10 in [48] is highly recommended as a reference. The vbSPT algorithm [50] is a prototypical application of the approach to spaSPT data.

For practical spaSPT analysis, the major advantage of a full VB treatment over EM is that it provides a natural criterion for model selection - in particular, for choosing the number of diffusing states in a mixture. This criterion appears in the form of the *variational lower bound* on the marginal likelihood. (This quantity is also sometimes known as the evidence lower bound or "ELBO".) Because VB maximizes the variational lower bound rather than the model likelihood, it incorporates a natural penalty on model complexity in the form of the negative entropies of the posterior distribution. As a result, the framework naturally favors the simplest possible models that describe the data.

In this section, first we provide a brief review of VB methods in the context of spaSPT, then proceed to derive what is probably the simplest possible variational framework for mixtures of regular Brownian motions. Next, we show how this framework can be used for model selection. Finally, we comment on some of the limitations of variational Bayes methods for spaSPT analysis, particularly when applied to non-normal diffusion models.

5.2.1 Variational lower bound

For a second, we'll forget about trajectories and diffusion coefficients to keep the notation less cluttered. Suppose that $\mathbf{X}$ represents a set of random variables that we have observed and $\mathbf{Y}$ represents a set of random variables that we have *not* observed. For example, $\mathbf{X}$ might be some data and $\mathbf{Y}$ might be model parameters or state assignments for each data point. From a Bayesian context, there is no fundamental difference between $\mathbf{X}$ and $\mathbf{Y}$ apart from that one is observed and one is not. In other words, both are treated as sets of random variables that are linked by some kind of relationship (a "model").

Our goal is to see how much we can learn about $\mathbf{Y}$ given the observed $\mathbf{X}$. The recipe to do this is Bayes' theorem:

$$p\left(\mathbf{Y}|\mathbf{X}\right) = \frac{p\left(\mathbf{X}|\mathbf{Y}\right)p\left(\mathbf{Y}\right)}{p\left(\mathbf{X}\right)}$$

For the vast majority of models, one or more of these terms will be analytically intractable. In particular, the *marginal likelihood* (also known as the *model evidence*)

$$p(\mathbf{X}) = \int_{\mathbf{Y}} p(\mathbf{X}, \mathbf{Y}) \mathrm{d}\mathbf{Y}$$

lacks a closed-form expression for all but the simplest models. All of the Bayesian methods considered in this thesis are ways to work around this problem. They propose some kind of approximation to the posterior that is refined through iteration, often without having to evaluate $p(\mathbf{X})$. In the case of Monte Carlo techniques like Gibbs sampling, the approximation is numerical, achieved by simulation. In variational Bayes, the approximation is some analytical function $q(\mathbf{Y})$. Our goal is to change q until

$$q(\mathbf{Y}) \approx p(\mathbf{Y}|\mathbf{X})$$

In other words, we want our inference method to find a $q(\mathbf{Y})$ that is in some sense "close" to the real posterior. A natural way to quantify "closeness" is the Kullback-Leibler divergence

$$\mathrm{KL}(q||p) = -\int_{\mathbf{Y}} q(\mathbf{Y}) \log\left[\frac{p(\mathbf{Y}|\mathbf{X})}{q(\mathbf{Y})}\right] \mathrm{d}\mathbf{Y}$$

(The integral can be replaced with summation, for parts of $\mathbf{Y}$ that are discrete.) An important property of the divergence is that $\mathrm{KL}(q||p) \geq 0$, with equality holding only if q and p are identical. As a result, one way to obtain an approximation to the posterior is to minimize the Kullback-Leibler divergence with respect to q. The problem is that the divergence is difficult to minimize directly: after all, it depends on $p(\mathbf{Y}|\mathbf{X})$, which is exactly what we're trying to infer.

There is a way around this difficulty. Define the *variational lower bound*

$$L[q] = \int_{\mathbf{Y}} q(\mathbf{Y}) \log \left[\frac{p(\mathbf{X}, \mathbf{Y})}{q(\mathbf{Y})} \right] \, \mathrm{d}\mathbf{Y} \tag{5.2}$$

$L[q]$ is a functional, mapping each distribution q to a real number. The motivation for the name "lower bound" will become clear shortly.

According to the multiplication law of probability,

$$p(\mathbf{X}, \mathbf{Y}) = p(\mathbf{Y}|\mathbf{X})p(\mathbf{X})$$

Substituting this into 5.2 and rearranging, we obtain

$$\log p(\mathbf{X}) = L[q] + \mathrm{KL}(q||p)$$

Since $\mathrm{KL}(q||p) \geq 0$, we have $\log p(\mathbf{X}) \geq L[q]$ So $L[q]$ places a lower bound on the marginal log likelihood (hence its name). The two are equal if and only if the Kullback-Leibler divergence between $q(\mathbf{Y})$ and $p(\mathbf{Y}|\mathbf{X})$ is zero - that is, if our approximation is perfect. Since $p(\mathbf{X})$ doesn't depend on $\mathbf{Y}$, it is a constant with respect to our choice of q. As a result, minimizing $\mathrm{KL}(q||p)$ is equivalent to maximizing $L[q]$.

This is vital because it is usually easier to maximize $L[q]$ than it is to minimize $\mathrm{KL}(q||p)$. $L[q]$ depends on $\log p(\mathbf{X}, \mathbf{Y})$ (the "complete log likelihood" from the previous chapter) rather than the posterior $\log p(\mathbf{Z}|\mathbf{X})$. The former is often far easier to work with, as we saw in the discussion of finite-state mixture models in the previous section.

In order to maximize $L[q]$, we next inspect the structure of the approximative posterior q.

5.2.2 Factorable approximations to the posterior

So far, we've made no assumptions about the form of the model apart from segregating the random variables into two categories $\mathbf{X}$ and $\mathbf{Y}$. In particular, the form

of q remains completely unspecified.

One approach to define q is simply to choose some function expected to provide reasonably good approximations to the true posterior. While viable, this approach often introduces more assumptions about the form of the posterior than is strictly necessary. The alternative proposed in variational Bayes is the following.

Segregate the parameters $\mathbf{Y}$ into disjoint groups labeled by k, so that the k^{th} group is $\mathbf{Y}_k$. We assume that the function q is separable with respect to these groups:

$$q(\mathbf{Y}) = \prod_k q_k(\mathbf{Y}_k) \tag{5.3}$$

For many models, appropriate selection of the groups along with the priors and the likelihood $p(\mathbf{X}|\mathbf{Y})$ is sufficient to induce a closed form for the factors q_k. In other words, this is the only assumption we need to make.

To see this, we'll focus on a particular term $q_j(\mathbf{Y}_j)$, holding the others constant. Substituting 5.3 into 5.2, we have

$$\begin{aligned} L[q] = \int_{\mathbf{Y}_j} q_j(\mathbf{Y}_j) \left(\int_{\mathbf{Y}_{k\neq j}} \log p(\mathbf{X},\mathbf{Y}) \prod_{k\neq j} q_k(\mathbf{Y}_k) \mathrm{d}\mathbf{Y}_{k\neq j} \right) \mathrm{d}\mathbf{Y}_j \\ - \sum_k \int_{\mathbf{Y}_k} q_k(\mathbf{Y}_k) \log q_k(\mathbf{Y}_k) \mathrm{d}\mathbf{Y}_k \end{aligned}$$

The second term is just the sum of entropies of the individual factors q_k. This provides a hint of the natural penalty against model complexity built into variational Bayes - part of maximizing $L[q]$ involves maximizing the entropies of the factors of q. The first term is the negative "cross-entropy" between q_j and the term in parentheses, which is also a function of $\mathbf{Y}_j$. Call this function $\log \overline{p}$; that is, define

$$\log \overline{p}\left(\mathbf{X}, \mathbf{Y}_j\right) = \mathbb{E}_{\mathbf{Y}_{k\neq j}}\left[\log p(\mathbf{X},\mathbf{Y})\right] + \text{constant} \tag{5.4}$$

Here, the constant is chosen such that the distribution $\overline{p}(\mathbf{X}, \mathbf{Z}_j)$ is normalized. Then the lower bound can be expressed

$$\begin{aligned} L[q] &= \int_{\mathbf{Y}_j} q_j(\mathbf{Y}_j) \log \overline{p}\left(\mathbf{X}, \mathbf{Y}_j\right) \mathrm{d}\mathbf{Y}_j - \int_{\mathbf{Y}_j} q_j(\mathbf{Y}_j) \log q_j(\mathbf{Y}_j) \mathrm{d}\mathbf{Y}_j + \text{constant} \\ &= \int_{\mathbf{Y}_j} q_j(\mathbf{Y}_j) \log \left[\frac{\overline{p}(\mathbf{X}, \mathbf{Y}_j)}{q_j(\mathbf{Z}_j)} \right] + \text{constant} \\ &= -\mathrm{KL}(q_j || \overline{p}) + \text{constant} \end{aligned}$$

All terms that do not depend on $\mathbf{Y}_j$, including the entropies of the other $q_{k\neq j}$, have been absorbed into the constant. If we maximize $L[q]$ with respect to the term q_j without changing the other factors of q, then this constant is irrelevant. From this we see that $L[q]$ is maximized with respect to q_j when the Kullback-Leibler divergence $\mathrm{KL}(q_j||\overline{p})$ is minimized. Since the divergence is zero when its arguments are equal, this condition is just the condition $q_j(\mathbf{Y}_j) = \overline{p}(\mathbf{X}, \mathbf{Y}_j)$. So, using 5.4, the ideal choice for q_j is

$$\log q_j(\mathbf{Y}_j) = \mathbb{E}_{\mathbf{Z}_{k\neq j}}[\log \overline{p}(\mathbf{X}, \mathbf{Y}))] + \text{constant} \tag{5.5}$$

Equations 5.3 and 5.5 represent the central machinery of the variational Bayes approach. The constant in 5.5 should be chosen so that the distribution q_j is normalized. When working with conjugate priors, this is usually automatically determined by finding $\mathbb{E}_{\mathbf{Y}_{k\neq j}}[\log \overline{p}]$.

In the expectation 5.5, the optimal form for q_j is conditional on the other $q_{k\neq j}$. So in practice we cycle between the different q_j, deriving each in turn until some convergence criterion is reached. Procedurally:

1. Choose some initial guess for each q_j.
2. For iterations $t = 1, 2, ...$:
   (a) For each q_j, hold $q_{k\neq j}$ constant and find q_j such that
   $$\log q_j(\mathbf{Y}_j) = \mathbb{E}_{\mathbf{Y}_{k\neq j}}[\log \overline{p}(\mathbf{X}, \mathbf{Y})] + \text{constant}$$
   (b) Call convergence either based on some statistic on the individual q_j or based on the lower bound $L[q]$.

5.2.3 Regular Brownian mixtures

Here, we apply the variational Bayes framework introduced in the previous section to the specific problem of mixtures of regular Brownian states. This is probably the simplest variational framework for spaSPT data. The version considered in [50] works with a different model that considers state transitions as well. Since typical trajectories in spaSPT data are very short, comprising only a handful of observations over several milliseconds, state transitions are difficult to infer with any degree of accuracy unless the datasets are very large and the model perfectly matches the data. So we neglect them. The framework presented here has the virtue of mathematical and computational simplicity.

We will assume that we have a dataset of N trajectories in m spatial dimensions. Use $\mathbb{X}$ to denote this set of trajectories. The trajectories have been tracked with

frame interval Δt and we assume that the localization error σ^2_{loc} is known and can be treated as a constant. (Later on, we'll examine the effect of variable localization error on inference.)

If our motion is regular Brownian with diffusion coefficient D, then each jump along a given spatial dimension has a normal distribution with variance $2(D\Delta t + \sigma^2_{\text{loc}})$. As a result, if the i^{th} trajectory has L_i jumps, then the sum of its squared displacements S_i has the distribution

$$S_i \sim \text{Gamma}\left(\frac{mL_i}{2}, \left(4(D\Delta t + \sigma^2_{\text{loc}})\right)^{-1}\right)$$

We will assume that the trajectory can inhabit any of K different diffusive states. To represent the state assignments, define a binary matrix $\mathbf{Z}$ of shape $N \times K$ so that z_{ij} is 1 if trajectory i belongs to state j and 0 otherwise. As a result, each row of $\mathbf{Z}$ sums to 1. Notice that this is a slightly different representation of $\mathbf{Z}$ than the one considered in the previous chapter; however, the spirit (assigning each trajectory to one of K states) is the same.

Define $\phi_j = 4(D_j\Delta t + \sigma^2_{\text{loc}})$, so that we have the single-trajectory likelihood

$$p(S_i \mid \text{state } j, \phi_j) = \text{Gamma}\left(\frac{mL_i}{2}, \phi_j^{-1}\right)$$

As a prior on each ϕ_j, choose $\phi_j \sim \text{InvGamma}\,(\alpha_0, \beta_0)$ so that

$$p(\phi_j) = \frac{\beta_0^{\alpha_0} e^{-\beta_0/\phi_j}}{\Gamma(\alpha_0)\phi_j^{\alpha_0+1}}$$

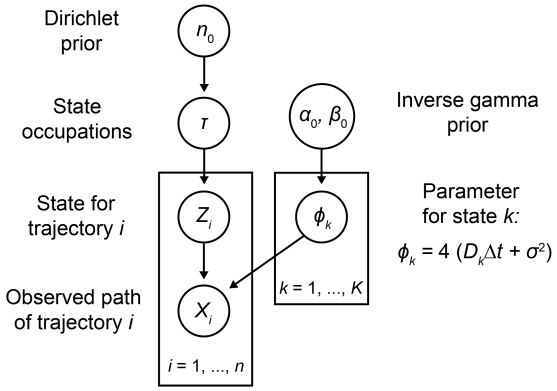

Figure 5.1: Graphical model for the variational Bayes model for regular Brownian motion considered in the text. Notice that rather than dealing with each state's diffusion coefficient D_k directly, we instead infer the "spatial variance" $\phi_k = 4(D_k\Delta t + \sigma^2_{\text{loc}})$, and then account for the effect of localization error later.

As a consequence, the posterior distribution over ϕ_j given a single trajectory in state j is

$$\phi_j \mid S_i \sim \text{InvGamma}\left(\alpha_0 + \frac{mL_i}{2}, \beta_0 + S_i\right)$$

We'll assume that the state occupations are $\boldsymbol{\tau}$, so that

$$p(\mathbf{Z}|\boldsymbol{\tau}) = \prod_{j=1}^{K}\prod_{i=1}^{N}\tau_j^{z_{ij}}$$

and we have the whole-dataset likelihood

$$p\left(\mathbb{X}|\mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi}\right) = \prod_{j=1}^{K}\prod_{i=1}^{N} p\left(S_i \mid \text{state } j, \phi_j^{-1}\right)^{z_{ij}}$$

Define a Dirichlet prior over the state occupations

$$\boldsymbol{\tau} \sim \text{Dirichlet}\left(n_0, ..., n_0\right) = \frac{1}{\mathbf{B}(n_0, ..., n_0)}\prod_{j=1}^{K}\tau_j^{n_0-1}$$

$\mathbf{B}(...)$ is the variadic beta function, defined by

$$\mathbf{B}(x_1, ..., x_K) = \frac{\Gamma(x_1)\cdot ... \cdot \Gamma(x_K)}{\Gamma(x_1 + ... + x_K)}$$

Notice that, if we were able to observe $\mathbf{Z}$, the posterior over $\boldsymbol{\tau}$ given $\mathbf{Z}$ would be

$$\boldsymbol{\tau} \mid \mathbf{Z} \sim \text{Dirichlet}\left(n_0 + \sum_{i=1}^{N} z_{i,0}, ..., n_0 + \sum_{i=1}^{N} z_{i,K}\right)$$

The choice of these conjugate priors over $\boldsymbol{\tau}$ and $\boldsymbol{\phi}$ will considerably simplify later steps.

Finally, note that the joint distribution of all variables factors as

$$p\left(\mathbb{X}, \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi}\right) = p\left(\mathbb{X} \mid \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi}\right) p\left(\mathbf{Z} \mid \boldsymbol{\tau}\right) p\left(\boldsymbol{\tau}\right) p\left(\boldsymbol{\phi}\right) \tag{5.6}$$

We seek an approximation $q(\mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi})$ to the posterior distribution $p(\mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi} \mid \mathbb{X})$. We make the mean-field approximation

$$q(\mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi}) = q(\mathbf{Z})q(\boldsymbol{\tau}, \boldsymbol{\phi}) \tag{5.7}$$

This is the sole approximation we need to make in order to obtain a tractable q. As we will see, this choice also induces further factoring of $q(\boldsymbol{\tau}, \boldsymbol{\phi})$ down the line.

In order to find the factors $q(\mathbf{Z})$ and $q(\boldsymbol{\tau}, \boldsymbol{\phi})$, we rely on the following iterative scheme:

1. Set $q^{(0)}(\mathbf{Z})$ and $q^{(0)}(\boldsymbol{\tau}, \boldsymbol{\phi})$ equal to the corresponding priors.
2. For each iteration $t = 1, 2, ...$:
   (a) Holding $q^{(t-1)}(\mathbf{Z})$ constant, find $q(\boldsymbol{\tau}, \boldsymbol{\phi})$ by solving
   $$\log q(\boldsymbol{\tau}, \boldsymbol{\phi}) = \mathbb{E}_{\mathbf{Z}} \left[\log p(\mathbb{X}, \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi})\right] + \text{constant}$$
   (b) Holding $q^{(t)}(\boldsymbol{\tau}, \boldsymbol{\phi})$ constant, find $q(\mathbf{Z})$ by solving
   $$\log q(\mathbf{Z}) = \mathbb{E}_{\boldsymbol{\tau}, \boldsymbol{\phi}} \left[\log p(\mathbb{X}, \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi})\right] + \text{constant}$$

First we focus on $q(\boldsymbol{\tau}, \boldsymbol{\phi})$. Substituting 5.6 into 5.5,

$$\begin{aligned}\log q(\boldsymbol{\tau}, \boldsymbol{\phi}) &= \mathbb{E}_{\mathbf{Z}} \left[\log p(\mathbb{X}|\mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi}) + \log p(\mathbf{Z}|\boldsymbol{\tau}) + \log p(\boldsymbol{\tau}) + \log p(\boldsymbol{\phi})\right] + \text{constant} \\ &= \sum_{j=1}^{K} \left(n_0 - 1 + \sum_{i=1}^{N} \mathbb{E}_{\mathbf{Z}} \left[z_{ij}\right] \right) \log \tau_j \\ &\quad + \sum_{j=1}^{K} \left[\left(\beta_0 + \sum_{i=1}^{N} \mathbb{E}_{\mathbf{Z}} \left[z_{ij}\right] S_i \right) \phi_j^{-1} - \left(\alpha_0 + 1 + \sum_{i=1}^{N} \mathbb{E}_{\mathbf{Z}} \left[z_{ij}\right] \right) \log \phi_j \right] + \text{constant}\end{aligned}$$

We have absorbed all terms that don't depend on $\boldsymbol{\tau}$ or $\boldsymbol{\phi}$ into the constant. This equation is additively separable in $\boldsymbol{\tau}$ and each ϕ_j, which implies that q further factors as

$$q(\mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi}) = q(\mathbf{Z}) q(\boldsymbol{\tau}) \prod_{j=1}^{K} q(\phi_j)$$

It's important to mention that we've no additional assumptions beyond 5.7. The additional factors are a consequence of the structure of our model and prior selection.

This factorization means that we can first address $q(\boldsymbol{\tau})$ and then each $q(\phi_j)$ separately. Absorbing all but terms dependent on $\boldsymbol{\tau}$ into the constant,

$$\log q(\boldsymbol{\tau}) = \sum_{j=1}^{K} \left(n_0 - 1 + \sum_{i=1}^{N} \mathbb{E}_{\mathbf{Z}} \left[z_{ij}\right] \right) \log \tau_j + \text{constant}$$

Taking the exponent, we see that this is another Dirichlet distribution:

$$q(\boldsymbol{\tau}) = \text{Dirichlet} \left(n_0 + \sum_{i=1}^{N} \mathbb{E}_{\mathbf{Z}} \left[z_{i,0}\right], ..., n_0 + \sum_{i=1}^{N} \mathbb{E}_{\mathbf{Z}} \left[z_{i,K}\right] \right)$$

Now, look closely at the terms $\sum_i \mathbb{E}_{\mathbf{Z}} \left[z_{ij}\right]$. Summing over the columns of $\mathbf{Z}$ gives us the *number of trajectories* assigned to each state. We might instead consider

counting the *number of jumps* assigned to each state. When there is no reason why the number of jumps per trajectory should depend on ϕ_j, both choices are equivalent. However, in real settings with finite depth of field, weighting by the number of jumps is far preferable for the reasons outlined in the previous chapter. Acknowledging this, we choose

$$q(\boldsymbol{\tau}) = \text{Dirichlet}\left(n_0 + N_1, ..., n_0 + N_K\right) \tag{5.8}$$

with

$$N_j = \sum_{i=1}^{N} \frac{mL_i}{2} \mathbb{E}_{\mathbf{Z}}\left[z_{ij}\right]$$

Given some $q(\mathbf{Z})$, we can evaluate the expectations with respect to $\mathbf{Z}$. We derive a closed form for these expectations shortly.

Turning to $q(\phi_j)$, we drop all terms that do not directly depend on ϕ_j. This leaves

$$\log q(\phi_j) = \left(\beta_0 + \sum_{i=1}^{N} \mathbb{E}_{\mathbf{Z}}\left[z_{ij}\right] S_i\right) \phi_j^{-1} - \left(\alpha_0 + 1 + \sum_{i=1}^{N} \frac{mL_i}{2} \mathbb{E}_{\mathbf{Z}}\left[z_{ij}\right]\right) \log \phi_j + \text{constant}$$

This is another inverse gamma density:

$$q(\phi_j) = \text{InvGamma}\left(\alpha_0 + \sum_{i=1}^{N} \frac{mL_i}{2} \mathbb{E}_{\mathbf{Z}}\left[z_{ij}\right], \beta_0 + \sum_{i=1}^{N} \mathbb{E}_{\mathbf{Z}}\left[z_{ij}\right] S_i\right) \tag{5.9}$$

With both $q(\boldsymbol{\tau})$ and $q(\boldsymbol{\phi})$ specified, we turn to $q(\mathbf{Z})$. This is defined through

$$\begin{aligned}
\log q(\mathbf{Z}) &= \mathbb{E}_{\boldsymbol{\tau},\boldsymbol{\phi}}\left[p(\mathbb{X}, \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi})\right] + \text{constant} \\
&= \mathbb{E}_{\boldsymbol{\tau},\boldsymbol{\phi}}\left[\log p(\mathbb{X} \mid \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi}) + \log p(\mathbf{Z} \mid \boldsymbol{\tau}) + \log p(\boldsymbol{\tau}) + \sum_{j=1}^{K} \log p(\phi_j)\right] + \text{constant} \\
&= \sum_{j=1}^{K} \sum_{i=1}^{N} \left(\left(\frac{mL_i}{2} - 1\right) \log S_i - S_i \mathbb{E}_{\boldsymbol{\phi}}\left[\phi_j^{-1}\right] - \log \Gamma\left(\frac{mL_i}{2}\right) \right. \\
&\qquad \left. - \frac{mL_i}{2} \mathbb{E}_{\boldsymbol{\phi}}\left[\log \phi_j\right] + \mathbb{E}_{\boldsymbol{\tau}}\left[\log \tau_j\right] \right) z_{ij} + \text{constant}
\end{aligned}$$

Taking the exponent, we have

$$q(\mathbf{Z}) \propto \prod_{j=1}^{K} \prod_{i=1}^{N} \rho_{ij}^{z_{ij}}$$

with ρ_{ij} defined via

$$\log \rho_{ij} = \left(\frac{mL_i}{2} - 1\right) \log S_i - S_i \mathbb{E}_{\boldsymbol{\phi}} \left[\phi_j^{-1}\right] - \log \Gamma \left(\frac{mL_i}{2}\right) - \frac{mL_i}{2} \mathbb{E}_{\boldsymbol{\phi}} \left[\log \phi_j\right] + \mathbb{E}_{\boldsymbol{\tau}} \left[\log \tau_j\right]$$

We can normalize by recognizing that $\mathbf{Z}$ represents a probability distribution over the different states for each trajectory. So let

$$r_{ij} = \frac{\rho_{ij}}{\sum_{j=1}^{K} \rho_{ij}}$$

whereupon

$$q(\mathbf{Z}) = \prod_{j=1}^{K} \prod_{i=1}^{N} r_{ij}^{z_{ij}} \tag{5.10}$$

Together, eqs. 5.8, 5.9, and 5.10 provide us with a way to determine the distributions $q(\mathbf{Z})$, $q(\boldsymbol{\tau})$, and $q(\boldsymbol{\phi})$. Each of these distributions is defined in terms of the expectations over some of the others - in particular, we need $\mathbb{E}_{\mathbf{Z}} \left[z_{ij}\right]$, $\mathbb{E}_{\boldsymbol{\tau}} \left[\log \tau_j\right]$, $\mathbb{E}_{\boldsymbol{\phi}} \left[\phi_j^{-1}\right]$, and $\mathbb{E}_{\boldsymbol{\phi}} \left[\log \phi_j\right]$. For the most part, these are well known results for their respective distributions:

$$\begin{aligned} \mathbb{E}_{\mathbf{Z}} \left[z_{ij}\right] &= r_{ij} \\ \mathbb{E}_{\boldsymbol{\tau}} \left[\log \tau_j\right] &= \psi\left(n_0 + N_j\right) - \psi\left(\sum_{k=1}^{K} \left(n_0 + N_k\right)\right) \\ \mathbb{E}_{\boldsymbol{\phi}} \left[\phi_j^{-1}\right] &= \frac{A_j}{B_j} \\ \mathbb{E}_{\boldsymbol{\phi}} \left[\log \phi_j\right] &= \log B_j - \psi(A_j) \end{aligned} \tag{5.11}$$

where

$$\begin{aligned} N_j &= n_0 + \sum_{i=1}^{N} \frac{mL_i}{2} \mathbb{E}_{\mathbf{Z}} \left[z_{ij}\right] \\ A_j &= \alpha_0 + \sum_{i=1}^{N} \frac{mL_i}{2} \mathbb{E}_{\mathbf{Z}} \left[z_{ij}\right] \\ B_j &= \beta_0 + \sum_{i=1}^{N} S_i \mathbb{E}_{\mathbf{Z}} \left[z_{ij}\right] \end{aligned}$$

and ψ is the digamma function, defined by

$$\psi(x) = \frac{\mathrm{d}}{\mathrm{d}x} \log \Gamma(x)$$

Finally, with the form of q specified, we can now determine the variational lower bound $L[q]$ on the marginal likelihood $\log p(\mathbb{X})$. Expanding this in terms of the factorizations 5.6 and 5.7,

$$\begin{aligned} L[q] &= \int q(\mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi}) \log \left[\frac{p(\mathbb{X}, \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi})}{q(\mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi})} \right] \mathrm{d}\mathbf{Z}\, \mathrm{d}\boldsymbol{\tau}\, \mathrm{d}\boldsymbol{\phi} \\ &= \mathbb{E}\left[\log p(\mathbb{X} \mid \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi})\right] + \mathbb{E}\left[\log p(\mathbf{Z} \mid \boldsymbol{\tau})\right] + \mathbb{E}\left[\log p(\boldsymbol{\tau})\right] + \mathbb{E}\left[\log p(\boldsymbol{\phi})\right] \\ &\quad - \mathbb{E}\left[\log q(\mathbf{Z})\right] - \mathbb{E}\left[\log q(\boldsymbol{\tau})\right] - \mathbb{E}\left[\log q(\boldsymbol{\phi})\right] \end{aligned}$$

where the expectations are taken with respect to the posterior variational distribution $q(\mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi})$. The result is entirely in terms of the previously defined expectations for z_{ij}, $\log \tau_j$, ϕ_j^{-1}, and $\log \phi_j$.

We can account for defocalization in the usual way by taking

$$q(\boldsymbol{\tau}) = \text{Dirichlet}\left(n_0 + \frac{N_1}{\eta_1}, ..., n_0 + \frac{N_K}{\eta_K}\right)$$

where η_j is proportional to the probability that a trajectory with spatial variance ϕ_j defocalizes after one frame interval.

Usually we find it convenient to set $\alpha_0 = n_0$, so that n_0 represents the pseudocounts accorded to the priors over both state occupations $\boldsymbol{\tau}$ and the spatial variances $\boldsymbol{\phi}$.

Algorithm 5.1: Variational Bayes inference for mixtures of regular Brownian states ("VB algorithm")

Parameters: $\mathbb{X}$, an experimental set of trajectories; Δt, the frame interval; Δz, the focal depth; σ^2_{loc}, the 1D localization error; K, the number of states in the mixture; n_0, the pseudocounts in the prior; $\mathbf{D}$, guesses for the diffusion coefficient of each state.

Precompute:

- For each trajectory i, calculate the sum of squared jumps S_i and the number of jumps L_i. For the hyperparameter $\beta_{0,j}$ on each ϕ_j, set $\beta_{0,j} = n_0/(4(D_j\Delta t + \sigma^2_{\text{loc}}))$. Make an initial guess for r_{ij} by taking $r_{ij} \propto \exp\left(-\frac{S_i n_0}{\beta_{0,j}} - \frac{mL_i}{2}\log\beta_{0,j}\right)$ and normalizing over the different states for each trajectory, so that $\sum_{j=1}^{K} r_{ij} = 1$

Algorithm: For each iteration $t = 1, 2, ...$

1. Set $n_j = n_0 + \frac{1}{\eta_j}\sum_{i=1}^{N}\frac{mL_i}{2}r_{ij}$. Set $A_j = n_0 + \sum_{i=1}^{N}\frac{mL_i}{2}r_{ij}$. Set $B_j = \beta_{0,j} + \sum_{i=1}^{N} S_i r_{ij}$.
2. Evaluate $\mathbb{E}\left[\log\tau_j\right] = \psi(n_j) - \psi\left(\sum_{j=1}^{K} n_j\right)$, $\mathbb{E}\left[\phi_j^{-1}\right] = A_j/B_j$, and $\mathbb{E}\left[\log\phi_j\right] = \log B_j - \psi(A_j)$.
3. Set $r_{ij} \propto \exp\left(\mathbb{E}\left[\log\tau_j\right] - S_i\mathbb{E}\left[\phi_j^{-1}\right] - \frac{mL_i}{2}\mathbb{E}\left[\log\phi_j\right]\right)$, and normalize over the K different states for each i so that $\sum_{j=1}^{K} r_{ij} = 1$.

Return:

- The set of r_{ij}, n_j, A_j, and B_j, which characterize the posterior according to

$$q(\mathbf{Z}) = \prod_{j=1}^{K}\prod_{i=1}^{N} r_{ij}^{z_{ij}}$$
$$q(\boldsymbol{\tau}) = \text{Dirichlet}\left(n_1, ..., n_K\right)$$
$$q(\boldsymbol{\phi}) = \prod_{j=1}^{K}\text{InvGamma}\left(A_j, B_j\right)$$

Algorithm 5.1 describes the implementation of this simple variational scheme, where we have let $\alpha_0 = n_0$ and we determine the priors $\beta_{0,j}$ for each state j based on a user guess for the diffusion coefficient. In practice, setting $n_0 < 10$ means that this specific guess has very little influence on the eventual result of the algorithm, although it serves a vital role in regularizing the early steps.

Notice that the output is the set of parameters governing the posterior approximation $q(\mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi})$. Most of the time, it's also useful to provide the posterior mean over each parameter in this distribution. These posterior means are given by

$$\mathbb{E}\left[z_{ij}\right] = r_{ij} \qquad \text{(likelihood for trajectory } i \text{ to inhabit state } j\text{)}$$

$$\mathbb{E}\left[\tau_j\right] = \frac{n_j}{\sum_{k=1}^{K} n_k} \qquad \text{(occupancy of state } j\text{)}$$

$$\mathbb{E}\left[D_j\right] = \frac{B_j}{4\Delta t(A_j - 1)} - \frac{\sigma_{\text{loc}}^2}{\Delta t} \qquad \text{(diffusion coefficient of state } j\text{)}$$

5.2.4 Automatic relevance determination

One of the main advantages of Bayesian methods over other approaches to analyzing spaSPT data, such as jump histogram fitting, is that the former deals much more naturally with model complexity. When using complex mixture models with a high number of components K, Bayesian methods tend to drive most of the state coefficients to zero, favoring more compact models. This is a consequence of maximizing the model evidence $p(\mathbf{X})$ rather than the model likelihood $p(\mathbf{X}|\mathbf{Y})$; the former balances increases in model likelihood against increases in the model's descriptive repertoire, or the range of possible data that can be generated from it. This quality of Bayesian algorithms is sometimes called *automatic relevance determination* in the context of machine learning. In contrast, maximum likelihood or least-squares approaches (such as jump histogram fitting) will tend to exploit all of the degrees of freedom in the model.

To investigate this effect, we simulated an spaSPT experiment in a 10 μm $\times$ 10 μm $\times$ 10 μm nucleus with a 700 nm focal depth, 10 ms frame intervals, and normally distributed localization error along each dimension with standard deviation $\sigma_{\text{loc}}^2 =$ 30 nm. Trajectories were drawn from one of four states:

- 10% chance to have diffusion coefficient 0.001 μm^2 s^{-1}
- 30% chance to have diffusion coefficient 0.7 μm^2 s^{-1}
- 20% chance to have diffusion coefficient 2.3 μm^2 s^{-1}
- 40% chance to have diffusion coefficient 8.0 μm^2 s^{-1}

The same set of trajectories was used for fitting by two approaches - the variational Bayes approach (Algorithm 5.1) or a jump length histogram fitting approach ("LS", analogous to [60]). For each method, we tried fitting with mixtures of four, eight, or sixteen states ($K = 4, 8, 16$). Because only 5246 trajectories were observed in the focal volume, the information available to both methods is highly limited.

The results are summarized in the following table. The columns correspond to the outputs of either the VB or the least-squares jump histogram ("LS") algorithms.

$K = 4$	VB occupancy	VB diff. coef. (μm^2 s^{-1})	LS occupancy	LS diff. coeff. (μm^2 s^{-1})
State 1	0.115	0.008	0.108	0.000
State 2	0.364	0.802	0.370	0.753
State 3	0.157	2.739	0.176	2.873
State 4	0.363	7.931	0.346	8.429

$K = 8$	VB occupancy	VB diff. coef. (μm^2 s^{-1})	LS occupancy	LS diff. coef. (μm^2 s^{-1})
State 1	0.115	0.009	0.108	0.000
State 2	0.364	0.804	0.127	0.691
State 3	0.155	2.749	0.117	0.709
State 4	0.001	2.841	0.141	0.929
State 5	0.001	3.041	0.110	2.904
State 6	0.001	3.727	0.071	4.303
State 7	0.001	5.767	0.027	4.807
State 8	0.362	7.944	0.299	8.933

$K = 16$	VB occupancy	VB diff. coef. (μm^2 s^{-1})	LS occupancy	LS diff. coef. (μm^2 s^{-1})
State 1	0.115	0.008	0.033	0.000
State 2	0.362	0.801	0.029	0.000
State 3	0.001	2.521	0.045	0.000
State 4	0.027	2.697	0.110	0.701
State 5	0.057	2.726	0.151	0.711
State 6	0.001	2.730	0.021	0.721
State 7	0.070	2.731	0.033	0.911
State 8	0.001	2.756	0.054	1.004
State 9	0.001	3.062	0.000	1.330
State 10	0.001	3.441	0.049	1.924
State 11	0.001	4.619	0.061	2.731
State 12	0.001	5.241	0.015	4.336
State 13	0.001	5.411	0.057	5.185
State 14	0.001	5.613	0.056	5.523
State 15	0.001	5.690	0.114	7.012
State 16	0.360	7.945	0.171	10.110

Both the VB algorithm and the LS algorithm perform similarly when using $K = 4$ - which is the true number of underlying states. However, their behavior departs markedly when using higher K. The jump histogram fitting approach tends to use as many components as it can, so that at $K = 8$ or $K = 16$, we have a large number of low-occupancy states. In fact, we may make very different biological interpretations on the outcome of the method when $K = 4$ than when $K = 16$. In contrast, the outputs of the VB algorithm are much sparser, with most of the occupancies close to zero. In fact, if we ignore the components with occupancies

close to zero, the result when $K = 16$ is actually fairly similar to the result when $K = 4$. The only major difference is that one of the intermediate states has been split into three states with similar diffusion coefficients.

Fig. 5.2 systematically compares the two methods across a broader range of conditions. In general, the jump histogram fitting method takes advantage of the full set of states available to it, giving little indication of the true number of underlying states. In contrast, the VB method tends to approach a stable number of significantly occupied states that are close to the true number.

The ability of Bayesian methods to respond intelligently to changes in the complexity of the model via the "automatic relevance determination" effect is one of their advantages over the jump length histogram fitting or maximum likelihood approaches. The algorithm tends to generate the simplest possible models to describe the data and does not necessarily exploit every degree of freedom that

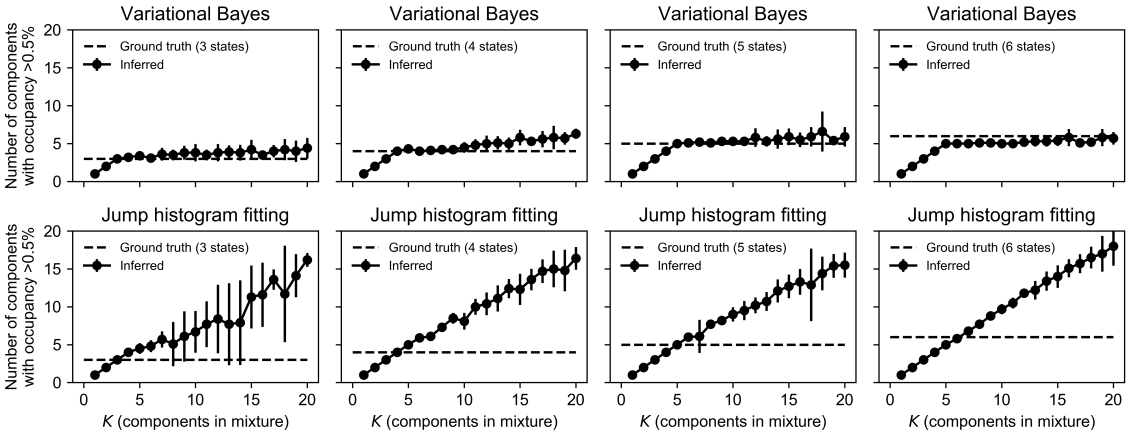

Figure 5.2: Systematic comparison of the ability of the VB algorithm 5.1 and jump histogram fitting to infer the number of components in a mixture of Brownian states. The *x*-axis of each plot represents the number of components that were allowed in the model fit, and the *y*-axis represents the number of those components with significant occupancy after fitting. The dotted lines represent the ground truth for the simulations. Trajectories with the indicated number of states were simulated in a nucleus with 5 μm radius, 700 nm focal depth, 10 ms frame intervals, 30 nm localization error, and a 20 Hz bleaching rate. Each data point represents the result of 5-10 simulations. Proceeding from left to right, the state parameters and occupancies were as follows: (*far-left*) diffusion coefficients 0.01, 1.5, 8.0 μm^2 s^{-1}, occupancies 0.1, 0.3, 0.6; (*mid-left*) diffusion coefficients 0.001, 0.7, 1.5, 8.0 μm^2 s^{-1}, occupancies 0.1, 0.3, 0.2, 0.4; (*mid-right*) diffusion coefficients 0.001, 0.3, 1.3, 3.5, 10.0 μm^2 s^{-1}, occupancies 0.1, 0.2, 0.4, 0.1, 0.2; (*far right*) diffusion coefficients 0.001, 0.2, 1.0, 2.5, 5.5, 10.0 μm^2 s^{-1}, occupancies 0.15, 0.1, 0.15, 0.3, 0.1, 0.2.

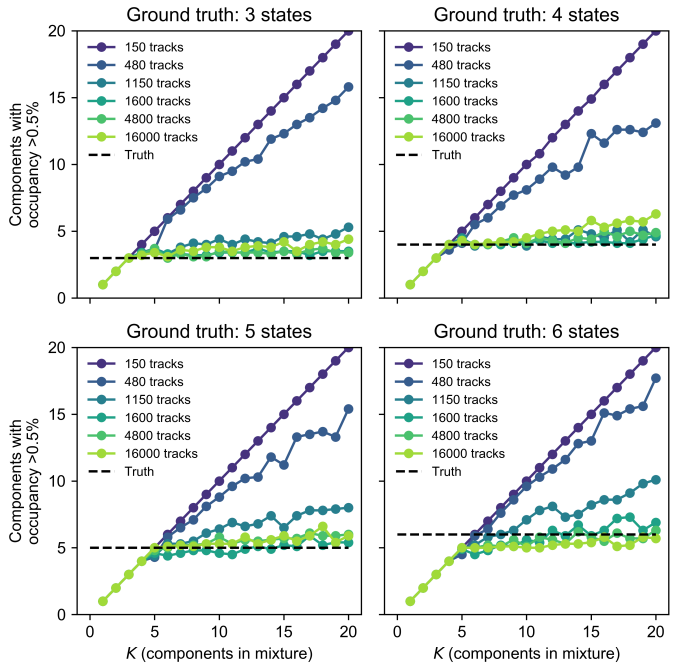

Figure 5.3: Comparison of the number of trajectories against the number of components with significant occupancy recovered by the variational Bayes algorithm. The simulations were performed the same as in 5.2. The dotted lines in each subplot represent the true number of states in the corresponding simulation.

we give it. This means that the outcome of our analysis with Bayesian methods like the VB algorithm is less sensitive on the parameters we use. Whereas changing K qualitatively changes the output of the jump histogram fitting approach, it has little effect on the VB algorithm.

A more sophisticated approach, explored by [84], explicitly incorporates the number of components K as a random variable in the model and mixes the models over K. This approach can be seen as an alternative to the Dirichlet processes considered later in the chapter.

Importantly, the ability of the VB algorithm to discern the true number of states is critically dependent on the amount of data available. In Fig. 5.3, we can see that when only a few hundred trajectories are used, the algorithm is not confident enough to drive most of the occupancies to zero. This is a direct consequence of the strength of the prior. As more trajectories are acquired, the relative weight of the data against the prior becomes stronger and the algorithm approaches the true number of underlying states in the mixture.

Fig. 5.4 demonstrates several runs of the VB algorithm on experimental SPT trajectories. In this plot, the area of each state is proportional to the posterior mean occupation of that state. For instance, the posterior occupations of the fastest diffusing state for HT and HT-NLS account for about 70% and 85% of the total occupation for those conditions, respectively.

Notice that after about 6 states, we don't acquire much additional posterior model complexity by adding more states to the model. This is a general result for the VB algorithm; unless there is very strong evidence for more states, usually it converges on 5-6 posterior states with significant occupation.

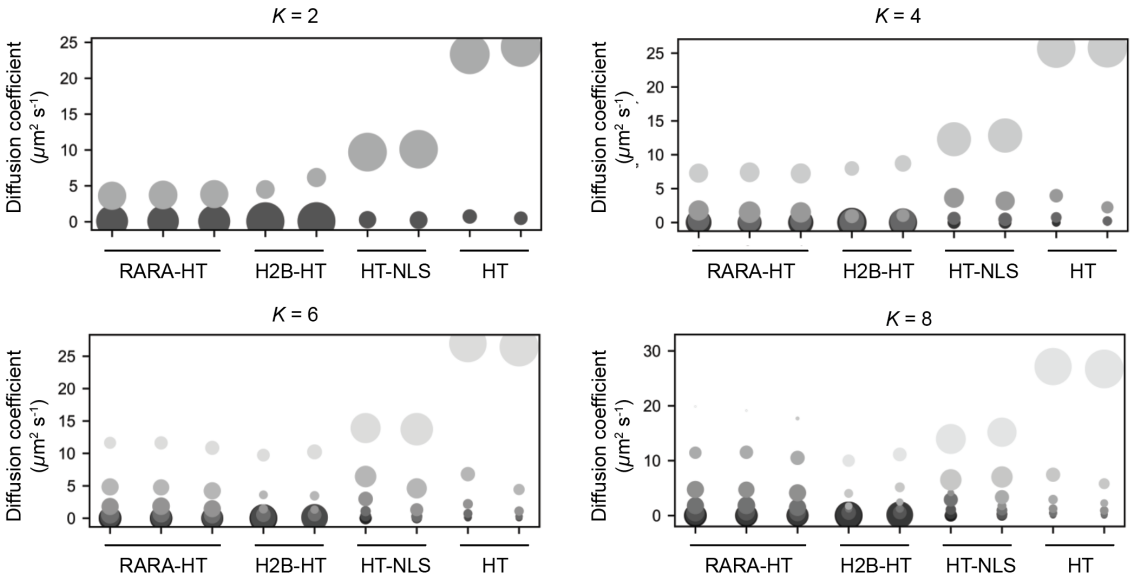

Figure 5.4: Posterior models for the VB algorithm (Algorithm 5.1) on experimental SPT trajectories. Each point is a state, and the area of each point is proportional to the occupation of that state under the posterior model. *K* reflects the number of states allowed during fitting. The *x*-labels have the following meaning: *RARA-HT*, endogenously tagged retinoic acid receptor α-HaloTag-3xFLAG in U2OS nuclei; *H2B-HT*, stably transfected H2B-HaloTag-SNAPf in U2OS nuclei; *HT*, transiently transfected HaloTag-MCS in U2OS nuclei; *HT-NLS*, transiently transfected HaloTag-3xNLS in U2OS nuclei. Cells were labeled with 100 nM PA-JFX549 for 30 min, followed by four 30 min washes. Tracking was performed with 7.48 ms frame intervals, 1.5 ms pulse widths on microscope with approximately 700 nm depth of field and 160 nm pixels; the approximate 1D dimensional root positional variance associated with localization under these settings is $\sim$ 35 nm. Each column represents a separate biological replicate. For the RARA-HT samples, each biological replicate was taken from a separate knock-in clone.

5.2.5 Comparison with EM

The variational Bayes method represented by Algorithm 5.1 is an expanded version of the EM algorithm outlined in the previous chapter. Recall that the EM algorithm worked by maximizing a "merit function". In that algorithm, we used θ_j to represent the parameters for the j^{th} diffusive state. (In the variational Bayes algorithm for RBMs, the only parameter in θ_j is the spatial variance ϕ_j.) The update criterion for $\boldsymbol{\tau}$ and $\boldsymbol{\theta}$ was

$$\begin{aligned}\boldsymbol{\tau}^{(t+1)}, \boldsymbol{\theta}^{(t+1)} &= \underset{\boldsymbol{\tau},\boldsymbol{\theta}}{\operatorname{argmax}}\, Q\left(\boldsymbol{\tau}, \boldsymbol{\theta} \mid \boldsymbol{\tau}^{(t)}, \boldsymbol{\theta}^{(t)}\right) \\ &= \underset{\boldsymbol{\tau},\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathbb{E}_{\mathbf{Z}|\mathbb{X},\boldsymbol{\tau}^{(t)},\boldsymbol{\theta}^{(t)}}\left[\log \mathcal{L}\left[\boldsymbol{\tau}, \boldsymbol{\theta} \mid \mathbb{X}, \mathbf{Z}\right]\right] \\ &= \underset{\boldsymbol{\tau},\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathbb{E}_{\mathbf{Z}|\mathbb{X},\boldsymbol{\tau}^{(t)},\boldsymbol{\theta}^{(t)}}\left[\log p\left(\mathbb{X}, \mathbf{Z} \mid \boldsymbol{\tau}, \boldsymbol{\theta}\right)\right]\end{aligned}$$

We saw that this solution could be expressed

$$\begin{aligned}\tau_j^{(t+1)} &= \sum_{i=1}^{N} T_{ji}^{(t)} \\ \theta_j^{(t+1)} &= \underset{\theta_j}{\operatorname{argmax}} \sum_{i=1}^{N} T_{ji}^{(t)} \log f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid \text{state } j, \theta_j\right)\end{aligned}$$

where the matrix $\mathbf{T}^{(t)}$ was defined in terms of the previous iteration's parameters

$$T_{ji}^{(t)} = \frac{\tau_j^{(t)} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid \text{state } j, \theta_j^{(t)}\right)}{\sum_{k=1}^{K} \tau_k^{(t)} f_{\mathbf{X}|Z}\left(\mathbf{X}_i \mid \text{state } j, \theta_k^{(t)}\right)}$$

Comparing the structure of this scheme with the update equations for $q(\mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi})$ in the variational Bayes algorithm, we immediately see that maximization of Q corresponds to taking

$$q\left(\boldsymbol{\tau}, \boldsymbol{\phi}\right) = \mathbb{E}_{\mathbf{Z}}\left[p\left(\mathbb{X}, \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\phi}\right)\right] + \text{constant}$$

In the EM algorithm, we held $\boldsymbol{\tau}$ and $\boldsymbol{\theta}$ constant at the previous iteration's values $\boldsymbol{\tau}^{(t)}$, $\boldsymbol{\theta}^{(t)}$. In the VB algorithm, we instead hold the *distributions* $q(\boldsymbol{\tau})$ and $q(\boldsymbol{\phi})$ constant. Meanwhile, taking $\mathbf{T}^{(t)}$ is analogous to evaluating the expectations $r_{ij} = \mathbb{E}_{\mathbf{Z}}\left[z_{ij}\right]$. Again, the only difference is that the VB algorithms works with a full approximation for the posterior, while EM only works with point estimates for the parameters (namely, their expectations with respect to the posterior).

If the EM and VB algorithms are so similar, what should compel us to use one over the other? While the EM algorithm can be slightly faster, there are two main reasons why the VB algorithm is superior:

- By providing a lower bound on the marginal likelihood, the VB algorithm provides a powerful criterion to choose between competing models for the same data. This is particularly useful when choosing the number of states K.
- The posterior estimate q can generate both point estimates for the model parameters in the style of EM, but can also be analyzed in its own right. For instance, we can determine whether any of the parameters differs substantially from its prior value, or use the entropies over each of the factors in q to determine their respective "information" contents.

However, VB has some drawbacks. These include:

- The accuracy with which q models the true posterior distribution is fundamentally dependent on the mean field approximations used to derive it.
- In mixture models, the identifiability problem means that q only models one of the $K!$ posterior maxima.
- The VB algorithm requires analytical solutions to 5.5 at each step. While we could obtain these for RBM, more general diffusion models (such as FBM) do not have analytical solutions without introducing further approximations.
- The VB algorithm does not handle non-discrete distributions of diffusion coefficients.

The last two points are the most limiting for us, and are the primary motivation for the next two classes of estimators considered in this chapter - arrayed state samplers and Dirichlet processes.

5.2.6 Accounting for localization error

A disadvantage of Algorithm 5.1 is that it only considers jumps over a single frame interval. This means that if the tracking algorithm produces jumps over gap frames, these jumps will not be used for inference, which is wasteful of data. Additionally, separating the spatial variance ϕ into temporally dependent and independent components is necessary to distinguish the contribution of the diffusion coefficient from the contribution of localization error, if the localization error is unknown.

We can make the following modifications to accommodate multiple frame intervals. Choose some maximum number of frame intervals to consider, and call this C. Instead of associating each trajectory with a single sum-squared jump S_i, instead associate it with a sequence of variables $(S_{i,c}, L_{i,c})$ for each frame gap being

considered ($c = 1, ..., C$). Here, $S_{i,c}$ is the sum of all squared jumps in that trajectory over c frame intervals, and $L_{i,c}$ is the number of such jumps.

For defocalization, rather than considering a single corrective factor η_j for state j, instead consider C different factors $\eta_{c,j}$. Each of these is defined as the probability that a particle with the spatial variance ϕ_j remains in the focal volume after c frames. These factors can be calculated at each iteration treating ϕ_j as if it were a constant.

Redefine the approximative posterior over the state occupations $\boldsymbol{\tau}$ as

$$
\begin{aligned}
q(\boldsymbol{\tau}) &= \text{Dirichlet}\left(n_1, ..., n_K\right) \\
n_j &= n_0 + \sum_{c=1}^{C} \left(\frac{1}{\eta_{c,\phi_j}} \sum_{i=1}^{N} r_{ij} \frac{m L_{i,c}}{2} \right)
\end{aligned}
$$

Further, redefine the approximative posterior over the spatial variance ϕ_j as

$$
\begin{aligned}
q(\phi_j) &= \text{InvGamma}\left(A_j, B_j\right) \\
A_j &= n_0 + \sum_{c=1}^{C} A_{j,c} \\
B_j &= \beta_0 + \sum_{c=1}^{C} B_{j,c} \\
A_{j,c} &= \sum_{i=1}^{N} r_{ij} \frac{m L_{i,c}}{2} \\
B_{j,c} &= \sum_{i=1}^{N} r_{ij} \left(\frac{S_{i,c} + 2 m L_{i,c} (c-1) \sigma_{\text{loc}}^2}{c} \right)
\end{aligned}
$$

The approximative posterior over $\mathbf{Z}$ then becomes

$$
\begin{aligned}
q(\mathbf{Z}) &= \prod_{j=1}^{K} \prod_{i=1}^{N} r_{ij}^{z_{ij}} \\
r_{ij} &= \frac{\rho_{ij}}{\sum_{k=1}^{K} \rho_{ik}} \\
\log \rho_{ij} &= \mathbb{E}_{\boldsymbol{\tau}} \left[\log \tau_j \right] - \sum_{c=1}^{C} \left[\frac{S_{i,c} A_{j,c}}{B_{j,c}} + \frac{m L_{i,c}}{2} \left(\log B_{j,c} - \psi(A_{j,c}) \right) \right]
\end{aligned}
$$

where ψ is the digamma function. We have the expected log occupancy $\mathbb{E}_{\boldsymbol{\tau}}\left[\tau_j\right] = \psi(n_j) - \psi\left(\sum_{k=1}^{K} n_k\right)$, as usual.

Notice that the priors over $\mathbf{Z}$, $\boldsymbol{\tau}$, and $\boldsymbol{\phi}$ are the same as before, but our likelihood has changed to a product of gamma distributions for each $(S_{i,c}, L_{i,c})$ with the variance parameters $\phi_{j,c} = c\phi_j - 4(c-1)\sigma^2_{\text{loc}}$.

Steps to update $q(\mathbf{Z})$, $q(\boldsymbol{\tau})$, and $q(\phi_j)$ can be interleaved with a update equation for σ^2_{loc}:

$$\sigma^2_{\text{loc}} \approx \frac{1}{\overline{N}} \sum_{c=2}^{C} \sum_{i=1}^{N} \left(\frac{\left(c\frac{mL_{i,c}}{2}\right)\phi_j - S_{i,c}}{4(c-1)} \right)$$

$$\overline{N} = \sum_{c=2}^{N} \sum_{i=1}^{N} \mathbb{I}_{L_{i,c}>0}$$

If desired, the single value for σ^2_{loc} can be replaced with multiple values $\sigma^2_{\text{loc},j}$ for each state j.

While the ability to model multiple frame intervals is useful when tracking with gaps, in general the assumed value of localization error does not have much of an effect on the regular finite-state variational Bayes algorithm (Fig. 5.5). As a result, it is generally recommended to hold localization error constant.

5.3 Arrayed state samplers

The variational Bayes (VB) methods considered in the previous chapter exhibit a remarkable ability to identify the number of diffusing states in mixtures. The fact that they work with analytical approximations to the posterior distribution means that we have access to a broad range of powerful analytical tools.

However, the VB method generalizes poorly to two important cases: (1) non-normal diffusive states and (2) non-discrete diffusive states.

What do we mean by "non-discrete diffusive states"? All of the models considered so far in this thesis could be described by the following generative model:

1. Choose a random state from a mixture of K different states. Each state j has some associated probability τ_j to be chosen.
2. Generate a trajectory based on the corresponding state parameters θ_j.

In this kind of scheme, the goal is to infer τ_j and θ_j for each state.

In cells, a given protein may participate in dozens of different complexes, each of which may have distinct dynamics in different environments. For instance, the TBP-associated factors (TAFs) that make up the TFIID complex may also participate in a variety of other complexes corresponding to variant TFIIDs, incomplete complexes, and monomers. Each of these states may have different diffusion coefficients depending on whether they are detected in the nucleoplasm, nucleoli, cytoplasm, or other compartments. As a result, there is strong motivation to question the assumption of a small number of discrete diffusing states, and develop a more general kind of mixture.

5.3.1 Principle

We can imagine generalizing the generative scheme above by eliminating the mixing coefficients τ, and instead sampling directly from a distribution over the state parameters:

1. Choose random state parameters θ from a distribution $f(\theta)$, which need not necessarily be discrete.
2. Generate a trajectory based on the chosen θ.

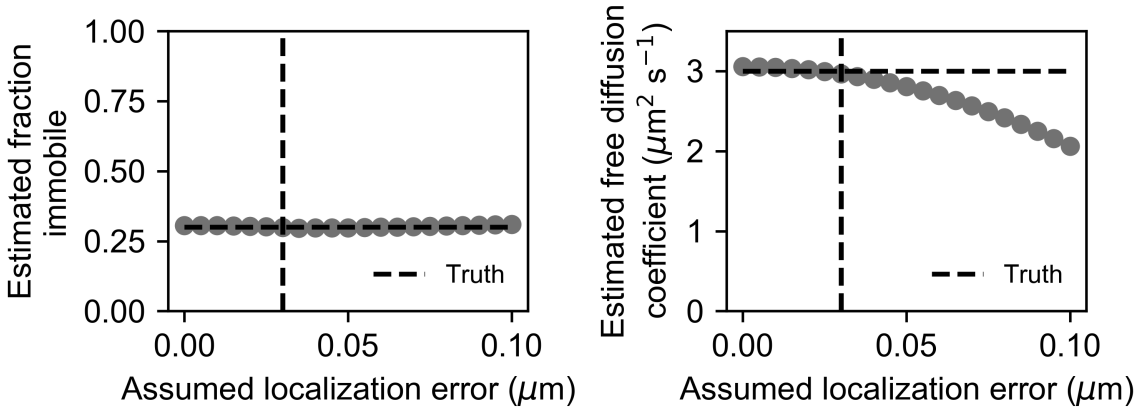

Figure 5.5: Sequential runs of the variational Bayes algorithm 5.1 on the same trajectories with different assumed localization error. Two regular Brownian states were simulated in a spherical nucleus with 5 μm radius, 700 nm focal depth, 10 ms frame intervals, 20 Hz bleaching rate, and 30 nm of normally distributed localization error along each axis. Trajectories were drawn from a slow state (0.001 μm^2 s^{-1}) with probability 0.3 and from a fast state (3.0 μm^2 s^{-1}) with probability 0.7.

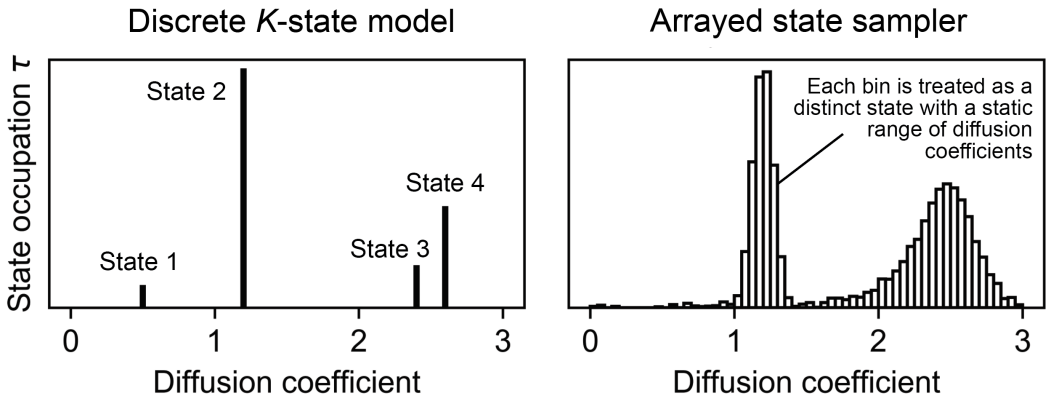

Figure 5.6: Schematic of the approach used by the arrayed state sampler for a regular Brownian mixture, compared to a discrete mixture model. Whereas a discrete mixture concentrates all of the probability density into a small number of diffusion coefficients, the arrayed state sampler instead partitions the range of possible diffusion coefficients into "bins", each of which is treated as a distinct state. This reduces the problem of model inference into estimating the posterior probability for each of the bins.

Our inference goal is then to find the distribution $f(\theta)$, rather than the values of τ_j and θ_j for each state.

How can we find $f(\theta)$? In the remainder of this chapter, we consider two approaches - arrayed state samplers and Dirichlet processes. In this section, we describe arrayed state samplers, which can be considered as a stepping stone to Dirichlet processes.

The idea of arrayed state samplers is schematized in Fig. 5.6. Rather than treating each of the diffusive states as discrete point densities in the space of model parameters, instead the range of possible model parameters is partitioned into bins, each of which is treated as a state with some occupation τ_k. If a trajectory is drawn from bin k, its parameters are assumed to be sampled from that bin with uniform probability density.

Similar to the VB algorithm, we'll treat the problem in a Bayesian manner, as illustrated in the graphical model in Fig. 5.7. For the prior over the state occupations $\boldsymbol{\tau}$, we take

$$\boldsymbol{\tau} \sim \text{Dirichlet}\,(n_0, ..., n_0)$$

We have seen that, given a trajectory with L_i jumps in m spatial dimensions with

a sum of squared jumps S_i, the likelihood of a particular diffusion coefficient D_j is

$$\mathcal{L}\left[\theta_j \mid \text{trajectory } i\right] = \text{Gamma}\left(\frac{mL_i}{2}, \frac{1}{4(D_j\Delta t + \sigma_{\text{loc}}^2)}\right)$$

where Δt is the frame interval and σ_{loc}^2 is the 1D variance due to localization error.

As in the case of the VB algorithm, it's convenient to work with the spatial variance $\phi = 4(D\Delta t + \sigma_{\text{loc}}^2)$ instead of D directly, so that the likelihood is just

$$\mathcal{L}\left[\theta_j \mid \text{trajectory } i\right] = \text{Gamma}\left(\frac{mL_i}{2}, \phi_j^{-1}\right)$$

In the arrayed state sampler, we need the likelihood of the k^{th} *bin* over ϕ rather than a single point. We take this to be the integrated likelihood between the bin edges ϕ_k and ϕ_{k+1}:

$$\begin{aligned}\mathcal{L}\left[\text{bin } k \mid \text{trajectory } i\right] &= \int_{\phi_k}^{\phi_{k+1}} \text{Gamma}\left(\frac{mL_i}{2}, \phi^{-1}\right) \mathrm{d}\phi \\ &= \frac{S_i^{\frac{mL_i}{2}-1}}{\Gamma\left(\frac{mL_i}{2}\right)} \int_{\phi_k}^{\phi_{k+1}} \phi^{-\frac{mL_i}{2}} e^{-S_i\phi^{-1}} \mathrm{d}\phi\end{aligned}$$

When $mL_i/2 > 1$, this can be expressed

$$\mathcal{L}\left[\text{bin } k \mid \text{trajectory } i\right] = \frac{\overline{\gamma}_I\left(\frac{mL_i}{2} - 1,\ S_i\phi_k^{-1}\right) - \overline{\gamma}_I\left(\frac{mL_i}{2} - 1,\ S_i\phi_{k+1}^{-1}\right)}{\Gamma\left(\frac{mL}{2} - 1\right)}$$

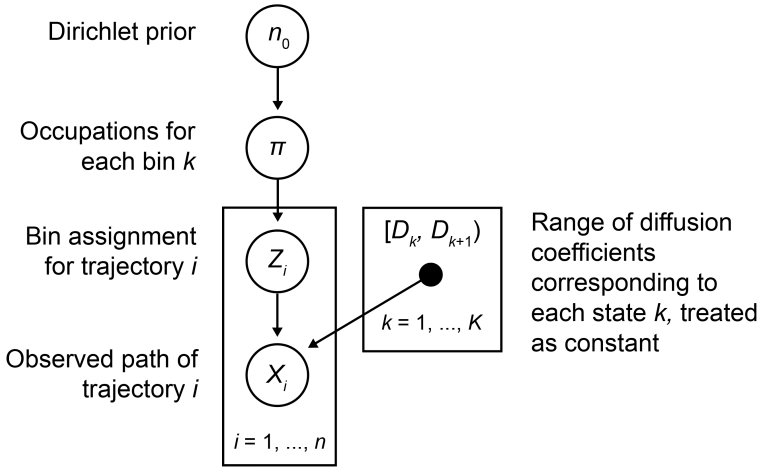

Figure 5.7: Graphical model for the arrayed state sampler. The solid black node corresponding to the range of state parameters $[D_k, D_{k+1})$ for state k indicates that this variable is treated as invariant.

where $\overline{\gamma}_l$ is the regularized lower incomplete gamma function

$$\overline{\gamma}_l(\alpha, x) = \frac{1}{\Gamma(\alpha)} \int_0^x t^{\alpha-1} e^{-t} \mathrm{d}t$$

When $mL_i/2 = 1$ - for instance, in the case of a trajectory with a single jump in two dimensions - $\overline{\gamma}_l$ is undefined and we must instead represent the likelihood as

$$\mathcal{L}[\text{bin } k \mid \text{trajectory } i] = \text{Ei}\left(-S_i \phi_k^{-1}\right) - \text{Ei}\left(-S_i \phi_{k+1}^{-1}\right)$$

where we've used the exponential integral

$$\text{Ei}(x) = -\int_{-x}^{\infty} t^{-1} e^{-t} \mathrm{d}t$$

Because the exponential integral diverges when $x \to 0$, it is necessary to constrain the inferred values of ϕ to lie below some maximum $\phi_{\text{max}} = 4(D_{\text{max}}\Delta t + \sigma_{\text{loc}}^2)$. We can also choose some appropriate ϕ_{min}, for instance by letting $D_{\text{min}} = 0$ whereupon $\phi_{\text{min}} = 4\sigma_{\text{loc}}^2$. Then the set of bins for ϕ can be obtained by partitioning the range $[\phi_{\text{min}}, \phi_{\text{max}}]$ into K bins.

5.3.2 Inference methods

Inference for arrayed state samplers can be accomplished in one of two ways:

1. Gibbs sampling. We start with some guess for $\mathbf{Z}^{(0)}$ and at each subsequent iterations $t = 1, 2, ...$ sample from the conditional distributions

$$\boldsymbol{\tau}^{(t)} \mid \mathbf{Z}^{(t-1)} \sim \text{Dirichlet}\left(n_0 + \sum_{i=1}^{N} \frac{mL_i}{2} z_{i,0}, ..., n_0 + \sum_{i=1}^{N} \frac{mL_i}{2} z_{i,K}^{(t-1)}\right)$$

$$\mathbf{Z}^{(t)} \mid \boldsymbol{\tau}^{(t)} \sim \prod_{j=1}^{K} \prod_{i=1}^{N} \left(\tau_j^{(t)}\right)^{z_{ij}}$$

2. Variational Bayes. We make the mean field approximation $p(\mathbf{Z}, \boldsymbol{\tau}) \approx q(\mathbf{Z})\, q(\boldsymbol{\tau})$, where $q(\mathbf{Z})$ is a categorical distribution and $q(\boldsymbol{\tau})$ is a Dirichlet distribution. Then we iteratively improve these distributions by taking

$$\log q(\mathbf{Z}) = \mathbb{E}_{\boldsymbol{\tau}}[\log p(\mathbb{X}, \mathbf{Z}, \boldsymbol{\tau})] + \text{constant}$$
$$\log q(\boldsymbol{\tau}) = \mathbb{E}_{\mathbf{Z}}[\log p(\mathbb{X}, \mathbf{Z}, \boldsymbol{\tau})] + \text{constant}$$

These approaches are outlined for the special case of regular Brownian motion in Algorithms 5.2 and 5.3. Both of these algorithms can be quite fast because the likelihood for each bin only needs to be calculated once, at the beginning of the algorithm.

Algorithm 5.2: Gibbs sampling for regular Brownian arrayed state samplers

Parameters: $\mathbb{X}$, an experimentally observed set of N trajectories in m spatial dimensions; Δt, the frame interval; σ^2_{loc}, the 1D localization error; D_{min} and D_{max}, the minimum and maximum diffusion coefficients to consider, and the number of bins K; n_0, the number of pseudocounts per bin in the prior.

Precompute:

- Calculate $\phi_{\text{min}} = 4(D_{\text{min}} + \sigma^2_{\text{loc}})$ and $\phi_{\text{max}} = 4(D_{\text{max}} + \sigma^2_{\text{loc}})$. Then partition the range $[\phi_{\text{min}}, \phi_{\text{max}}]$ into K bins, so that the k^{th} bin has edges (ϕ_k, ϕ_{k+1}).
- For each trajectory i, measure the number of jumps L_i and the sum of squared jumps S_i
- Calculate the likelihood matrix $\mathbf{A}$ where

$$A_{ji} = \begin{cases} \overline{\gamma}_I\left(\frac{mL_i}{2} - 1, \phi_k^{-1}\right) - \overline{\gamma}_I\left(\frac{mL_i}{2} - 1, \phi_{k+1}^{-1}\right) & \text{if } \frac{mL_i}{2} > 1 \\ \text{Ei}\left(-S_i\phi_k^{-1}\right) - \text{Ei}\left(-S_i\phi_{k+1}^{-1}\right) & \text{if } \frac{mL_i}{2} = 1 \end{cases}$$

- The initial binary state assignment matrix $\mathbf{Z}^{(0)}$, which has shape $K \times N$. For each trajectory i, set $Z_{ji}^{(0)} = 1$ with probability proportional to A_{ji} and set $Z_{ki}^{(0)} = 0$ for the other $k \neq j$.

Algorithm. For each iteration $t = 1, 2, ...$

1. For each bin j, calculate $N_j = \sum_{i=1}^{N} \frac{mL_i}{2} Z_{ji}^{(t-1)}$.
2. Sample $\boldsymbol{\tau}^{(t)} \sim \text{Dirichlet}\,(N_1, ..., N_K)$.
3. Sample $\mathbf{Z}^{(t)}$ conditional on $\boldsymbol{\tau}^{(t)}$ in the following way. For each trajectory i, choose a state j with probability

$$\frac{\tau_j^{(t)} A_{ji}}{\sum_{k=1}^{K} \tau_k^{(t)} A_{ki}}$$

   Set $Z_{ji}^{(t)} = 1$ for the selected j, and set $Z_{ki}^{(t)} = 0$ for all other $k \neq j$.

Return: The set of samples $\tau_j^{(t)}$; the posterior mean over the j^{th} bin is estimated $\frac{1}{T}\sum_t \tau_j$, where T is the total number of iterations

Algorithm 5.3: Variational Bayes estimation for regular Brownian arrayed state samplers

Parameters: $\mathbb{X}$, an experimentally observed set of N trajectories in m spatial dimensions; Δt, the frame interval; σ_{loc}^2, the 1D localization error; D_{min} and D_{max}, the minimum and maximum diffusion coefficients to consider, and the number of bins K; n_0, the number of pseudocounts per bin in the prior.

Precompute: $\phi_{\text{min}} = 4(D_{\text{min}} + \sigma_{\text{loc}}^2)$ and $\phi_{\text{max}} = 4(D_{\text{max}} + \sigma_{\text{loc}}^2)$. Then partition the range $[\phi_{\text{min}}, \phi_{\text{max}}]$ into K bins, so that the k^{th} bin has edges $(\phi_k,\ \phi_{k+1})$. For each trajectory i, measure the number of jumps L_i and the sum of squared jumps S_i. Then calculate the likelihood matrix $\mathbf{A}$ where

$$A_{ji} = \begin{cases} \frac{\bar{\gamma}_l\left(\frac{mL_i}{2}-1, \phi_k^{-1}\right) - \bar{\gamma}_l\left(\frac{mL_i}{2}-1, \phi_{k+1}^{-1}\right)}{\Gamma\left(\frac{mL_i}{2}-1\right)} & \text{if } \frac{mL_i}{2} > 1 \\ \text{Ei}\left(-S_i\phi_k^{-1}\right) - \text{Ei}\left(-S_i\phi_{k+1}^{-1}\right) & \text{if } \frac{mL_i}{2} = 1 \end{cases}$$

Create a $K \times N$ matrix $\mathbf{R}$ such that, initially, $R_{ji}^{(0)} = \frac{A_{ji}}{\sum_{k=1}^{K} A_{ki}}$.

Algorithm. For each iteration $t = 1, 2, ...$

1. For each bin j, calculate $n_j^{(t)} = \sum_{i=1}^{N} \frac{mL_i}{2} R_{ji}^{(t-1)}$. Then set

$$R_{ji}^{(t)} = \frac{A_{ji} e^{\psi(n_j^{(t)})}}{\sum_{k=1}^{K} A_{ki} e^{\psi(n_k^{(t)})}}$$

   where $\psi(x) = \frac{\mathrm{d}}{\mathrm{d}x} \log \Gamma(x)$ is the digamma function.

Return:

- Matrix $\mathbf{R}$ and vector $\mathbf{n}$, which characterize the approximation to the posterior distribution via

$$p(\mathbf{Z}, \boldsymbol{\tau} \mid \mathbb{X}) \approx q(\mathbf{Z})\, q(\boldsymbol{\tau})$$
$$q(\mathbf{Z}) = \prod_{j=1}^{K} \prod_{i=1}^{N} R_{ji}^{Z_{ij}}$$
$$q(\boldsymbol{\tau}) = \text{Dirichlet}(n_1, ..., n_K)$$

  The mean posterior occupation of each bin is $n_j / \sum_k n_k$.

In the case of the variational Bayes algorithm 5.3, we can also compute the evidence lower bound for the posterior distribution, which gives some insight into the arrayed state sampler. The complete likelihood factors as

$$p(\mathbb{X}, \mathbf{Z}, \boldsymbol{\tau}) = p(\mathbb{X} \mid \mathbf{Z}, \boldsymbol{\tau}) p(\mathbf{Z} \mid \boldsymbol{\tau}) p(\boldsymbol{\tau})$$

Then the lower bound is

$$\begin{aligned} L[q] &= \int_{\mathbf{Z},\boldsymbol{\tau}} q(\mathbf{Z}, \boldsymbol{\tau}) \log \left[\frac{p(\mathbb{X}, \mathbf{Z}, \boldsymbol{\tau})}{q(\mathbf{Z}) q(\boldsymbol{\tau})} \right] \mathrm{d}\mathbf{Z}\mathrm{d}\boldsymbol{\tau} \\ &= \mathbb{E}\left[\log p(\mathbb{X} \mid \mathbf{Z}, \boldsymbol{\tau})\right] + \mathbb{E}\left[\log p\left(\mathbf{Z} \mid \boldsymbol{\tau}\right)\right] + \mathbb{E}\left[p(\boldsymbol{\tau})\right] + H[q] \end{aligned}$$

where $H[q]$ is the entropy of q:

$$\begin{aligned} H[q] &= H[q(\mathbf{Z})] + H[q(\boldsymbol{\tau})] \\ &= -\int_{\mathbf{Z},\boldsymbol{\tau}} q(\mathbf{Z}, \boldsymbol{\tau}) \log q(\mathbf{Z}, \boldsymbol{\tau}) \mathrm{d}\mathbf{Z}\mathrm{d}\boldsymbol{\tau} \end{aligned}$$

The first term in $L[q]$ grows with the model likelihood $p(\mathbb{X}|\mathbf{Z}, \boldsymbol{\tau})$, the second and third terms are related to the priors, and the last grows with the entropy of the posterior approximation q. For a given choice of prior, the second and third terms are constant. So the algorithm boils down to balancing the model likelihood on one hand and the entropy of the posterior approximation on the other. Again, this demonstrates the tendency of the Bayesian approach to avoid overfitting by choosing simple models with a small number of components when possible.

5.3.3 Extension to mixtures of anomalous states

While arrayed state samplers have plenty of disadvantages - such as the assumption that a single likelihood characterizes each parameter bin - their major advantage is the ease with which they extend to non-normal modes of diffusion.

Fig. 5.8 demonstrates the extension of arrayed state samplers to resolve mixtures of fractional Brownian motions. Inference becomes increasingly challenging at higher numbers of states, and the dispersion of the parameters associated with any single state increases. Notice in particular that at six states, the fastest-diffusing state becomes joined into the second-fastest state.

5.4 Interpretation of aggregate likelihood methods

Together, the previous sections on finite-state variational Bayes and arrayed state samplers provide justification for the "aggregate likelihood" methods presented

in section 3.2. Here, we redevelop the aggregate likelihood method from the perspective of variational Bayes.

Suppose that we have a set of N trajectories, which we represent $\mathbb{X}$. The i^{th} trajectory is X_i. Suppose that there are L_i jumps in the trajectory i.

Just as for arrayed state samplers, we choose a set of K different diffusive states with fixed parameters θ_j for each $j \in \{1, ..., K\}$. The assignment of each trajectory to one of the K different states is represented by a matrix $\mathbf{Z}$ where $Z_{ij} = 1$ if trajectory i is from state j and 0 otherwise.

This situation is captured by the Bayesian mixture model

$$\begin{aligned} \boldsymbol{\tau} &\sim \text{Dirichlet}\,(n_0, ..., n_0) \\ Z_i &\sim \text{Mult}\,(\boldsymbol{\tau}, 1) \\ X_i \mid (i \text{ in state } j) &\sim f_{X|\theta}\left(X_i \mid \theta_j\right) \end{aligned}$$

Here, $f_{X|\theta}(X_i|\theta_j)$ is the likelihood function for our diffusion model, defined as the relative probability of seeing a trajectory X_i given that it comes from a state char-

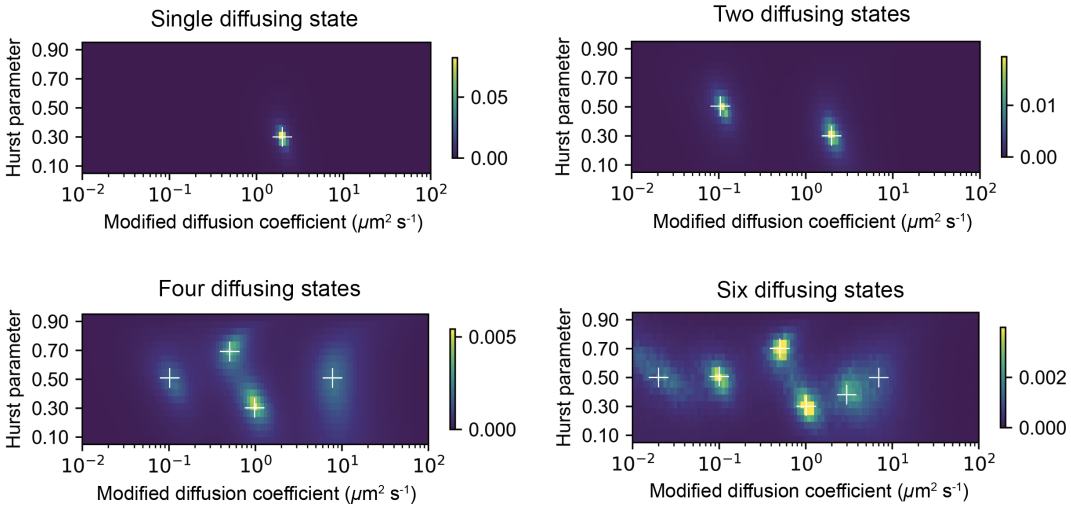

Figure 5.8: Application of arrayed state samplers to identify multiple anomalously diffusing states. Fractional Brownian motions (FBMs) were simulated in a spherical nucleus with 5 μm radius and 700 nm focal depth with a 10 Hz bleaching rate. The frame interval was 7.48 ms and 30 nm of normally-distributed 1D localization error was added to the coordinates of each point. Variable numbers of states were simulated; the white crosshairs indicate the true parameters for each state, while the color maps indicate the posterior state probabilities after running the arrayed state sampler algorithm.

acterized by the parameters θ_j.

The complete likelihood factors as

$$p(\mathbb{X}, \mathbf{Z}, \boldsymbol{\tau}) = p(\mathbb{X} \mid \mathbf{Z}, \boldsymbol{\tau})\, p(\mathbf{Z} \mid \boldsymbol{\tau})\, p(\boldsymbol{\tau})$$

We saw that if we approximate the posterior distribution of $\mathbf{Z}$ and $\boldsymbol{\tau}$ as $q(\mathbf{Z}, \boldsymbol{\tau}) = q(\mathbf{Z})\, q(\boldsymbol{\tau})$, then the choice for $q(\mathbf{Z})$ and $q(\boldsymbol{\tau})$ that maximizes the variational lower bound is

$$\begin{aligned} \log q(\mathbf{Z}) &= \mathbb{E}_{\boldsymbol{\tau}} [\log p(\mathbb{X}, \mathbf{Z}, \boldsymbol{\tau})] + \text{constant} \\ \log q(\boldsymbol{\tau}) &= \mathbb{E}_{\mathbf{Z}} [\log p(\mathbb{X}, \mathbf{Z}, \boldsymbol{\tau})] + \text{constant} \end{aligned}$$

Examining the first term,

$$\begin{aligned} \log q(\mathbf{Z}) &= \mathbb{E}_{\boldsymbol{\tau}} [\log p(\mathbb{X} \mid \mathbf{Z}) + \log p(\mathbf{Z} \mid \boldsymbol{\tau})] + \text{constant} \\ &= \sum_{j=1}^{K} \sum_{i=1}^{N} Z_{ij} \left(\log f_{X|\theta}(X_i \mid \theta_j) + \mathbb{E}_{\boldsymbol{\tau}} \left[\log \tau_j \right] \right) + \text{constant} \end{aligned}$$

Assume that $\mathbb{E}\left[\log \tau_j\right] = \text{constant}$ for all j. This is the case, for example, under the prior distribution for $\boldsymbol{\tau}$. Then, since $\sum_{j=1}^{K} Z_{ij} = 1$ for all i, the last term is absorbed into the constant and we have

$$\log q(\mathbf{Z}) = \sum_{j=1}^{K} \sum_{i=1}^{N} Z_{ij} \log f_{X|\theta}(X_i \mid \theta_j) + \text{constant}$$

Then the distribution for $q(\mathbf{Z})$ is

$$\begin{aligned} q(\mathbf{Z}) &= \prod_{i=1}^{N} \prod_{j=1}^{K} r_{ij}^{Z_{ij}} \\ r_{ij} &= \frac{f_{X|\theta}(X_i \mid \theta_j)}{\sum_{k=1}^{K} f_{X|\theta}(X_i \mid \theta_k)} \end{aligned}$$

Notice that this does not depend on $\boldsymbol{\tau}$. Now, treating $q(\mathbf{Z})$ as a fixed distribution, we have

$$\begin{aligned} q(\boldsymbol{\tau}) &= \frac{1}{B(n_1, \ldots, n_K)} \prod_{j=1}^{K} \tau_j^{n_j - 1} \\ n_j &= n_0 + \sum_{i=1}^{N} \frac{m L_i}{2} r_{ij} \end{aligned}$$

where L_i is the number of jumps in trajectory i. If we choose $n_0 = 0$, this is the aggregate likelihood function as defined in section 3.2. At this point, it should be clear that this is just the first iteration of a variational Bayes algorithm according to an arrayed state sampler. In other words, the aggregate likelihood is the mean estimate over $\boldsymbol{\tau}$ given a completely naive estimate for the distribution of the state assignments $\mathbf{Z}$. Importantly, this does not "mix" information between trajectories. It treats each trajectory as truly independent. While this is often less effective at picking out individual states, it is very unbiased and hence useful for nonparametric analyses of spaSPT data.

5.5 Dirichlet processes

With arrayed state samplers, we replaced point estimates with a distribution over state parameters like the diffusion coefficient. This distribution was discrete. It cut up the range of possible values for θ into bins, and then treated the likelihood of each bin as a constant.

A serious problem with this approach is that the selection of the bins is arbitrary. The algorithm's outcome may depend on whether the bins are coarse or fine. To see this, recall that one of the assumptions underying the arrayed state sampler for RBM is that if a trajectory is generated from a given bin, its diffusion coefficient is sampled from that bin with uniform probability density. This assumption made it possible to obtain an analytical expression for the integrated likelihood across the bin. But if the bins are *too* coarse, then our trajectory may be poorly described by the range of diffusion coefficients in that bin. Worse, it lends its inferential weight to trajectories that share little similarity with it. Finer bins, while more costly, are preferable. Exactly how fine is "fine enough" is not clear for a given diffusion model or dataset.

At this point, we should ask whether it is possible to let the width of the bins approach zero. In effect, this would replace the discrete distribution over the states with a continuous distribution. Rather than having a set of K bins, each with a range of parameter values $[\theta_k, \theta_{k+1}]$, we would have a single continuous distribution $f(\theta)$ so that the probability *density* to draw a trajectory with parameter θ is proportional to $f(\theta)$. The central challenge is that we can no longer rely on the discrete-state Dirichlet distribution to represent the state occupations. Computers cannot actually represent arbitrary continuous probability densities.

This challenge can be overcome with *Dirichlet processes*, which are infinite-dimensional analogs of the Dirichlet distribution. In this section, we first briefly review Dirichlet processes, then discuss their application to spaSPT data. As we will see, Dirichlet

processes lend themselves to a simpler Bayesian framework than the ones considered so far. We derive a Dirichlet process sampler specifically for RBM mixtures, and then explore its application to simulated and real datasets.

5.5.1 Summary of Dirichlet processes

Suppose that $\mathbb{X}$ denotes a set of trajectories that inhabit an unknown number of true diffusive states. Let's begin with a familiar approach: model this situation by a mixture of K different states, which are parametrized by one or more state variables θ_j for $j = 1, ..., K$. (For example, in a regular Brownian mixture, θ_j represents the diffusion coefficient.)

Each $X_i \in \mathbb{X}$ may be the raw spatial coordinates of the trajectory, the jump vectors, the sum of squared jumps, or whatever is convenient for the problem. Let $Z_i \in \{1, ..., K\}$ be the state assignment for the i^{th} trajectory.

A general Bayesian mixture model, such as the kind used in the finite-state VB algorithm, can be summarized as

$$\begin{aligned} \boldsymbol{\tau} &\sim \text{Dirichlet}\left(\frac{\alpha}{K}, ..., \frac{\alpha}{K}\right) \\ Z_i \mid \boldsymbol{\tau} &\sim \text{Mult}\left(\boldsymbol{\tau}, 1\right) \\ \theta_k \mid H &\sim H \\ X_i \mid Z_i, \boldsymbol{\theta} &\sim f_{X|\theta}(X_i \mid \theta_{Z_i}) \end{aligned} \tag{5.12}$$

Here, $\boldsymbol{\tau}$ is the vector of state occupancies, so that $\sum_{j=1}^{K} \tau_j = 1$. α is the "concentration parameter", to be investigated in detail in this section, and H is a prior distribution over each θ_k. $\text{Mult}\left(\boldsymbol{\tau}, 1\right)$ is a categorical random variable. It means that we drawn one of the K different states, with the probability to draw the j^{th} state given by τ_j. Finally, the last equation is the likelihood function - the probability to generate a trajectory, given a particular set of state parameters.

The central issue of this chapter is the selection of K, the number of states in the mixture. We want our model to be capable of capturing the complexity of multi-state diffusion in cells. But we don't want to overfit, creating a sea of meaningless parameters.

As we saw previously in the section on finite-state variational Bayes methods, increasing K means very different things depending on our analysis method. Maximum likelihood or least-squares algorithms like jump histogram fitting tend to exploit all of the degrees of freedom we give them. When K is too high, our

results are dominated by noise. While the resulting models appear to closely fit the training data, they generalize poorly outside of the experiment and tend to be uninterpretable. The variational Bayes (VB) algorithm behaved differently. In effect, the VB algorithm strikes a balance between how well the model describes the data (the model likelihood) and how probable the data is given the model (the model's descriptive repertoire). The latter condition emphasizes that in situations where many models can describe the data equally well, the simplest one is favored. When we specify values of K that are too high, Bayesian methods tend to drive most of the occupancies to zero, favoring simple models with the smallest number of components required to effectively describe the data.

For example, in Fig. 5.2, we saw that the VB algorithm recovered essentially the same model for a four-state mixture of Brownian states when $K = 4$, $K = 8$, and $K = 16$. But what happens when we make the number of components very high - say, $K = 100$ or $K = 1000$?

An excellent data-motivated investigation of this question is Radford Neal's 1992 chapter [85], which we strongly refer to the interested reader. As K becomes large, the number of components used to "explain" a given dataset of size N grows as $O(\alpha \log N)$, where α is the pseudocount parameter for the mixture model 5.12. (This relation is derived later in the section.) Importantly, this is independent of the number of components. In fact, we can even let $K \to \infty$ and the resulting models will still be discrete.

Allowing that $K \to \infty$ is the basis for the so-called "nonparametric Bayes" approach to mixture models. This approach has the benefit that it completely removes user decisions about the number of components in a mixture. The critical parameter that remains is α. What meaning does α have in this limit, and what exactly happens to the state occupancies?

Definition

The Dirichlet process (DP) can be seen as an infinite-dimensional generalization of the mixture model 5.12. As in the case of a regular Dirichlet distribution, draws from a DP are themselves probability distributions. However, these draws are different in character.

Suppose that the set of all possible state parameter values is Θ. For example, Θ may represent a range of biologically feasible diffusion coefficient - say, 0 to 100 $\mu m^2\ s^{-1}$. The measure G is distributed according to a Dirichlet process with concentration parameter α and base distribution H if, for any finite partition T_1,

..., $T_d \subseteq \Theta$, we have

$$(G(T_1), ..., G(T_d)) \sim \text{Dirichlet}\,(\alpha H(T_1), ..., \alpha H(T_d)) \tag{5.13}$$

That is, the marginal distributions of G are Dirichlet distributions. (For this expression to have any meaning, Θ must be G-measurable. Since this is the case for all of the diffusion models we deal with in spaSPT data, we generally gloss over issues of measurability in this thesis.)

Properties

An immediate consequence of the definition is that G is a probability measure, since the draws of the ordinary Dirichlet distribution are probability measures.

What do the parameters α and H mean? Taking the expectation of one of the elements in 5.13, which can be done easily using the properties of the regular Dirichlet distribution, we have

$$\mathbb{E}\left[G(T_j)\right] = \frac{H(T_j)}{\sum_k H(T_k)} = H(T_j)$$

The denominator is unity since H is a distribution on Θ. This makes it clear that the base distribution H plays the role of the "mean" for a DP. That is, a DP generates realizations distributed around H the same way that a regular random variable generates realizations distributed around its mean.

5.13 also illustrates the role of the concentration parameter. To see this, examine the variance of $G(T_j)$:

$$\text{Var}\left(G(T_j)\right) = \frac{H(T_j)\left(1 - H(T_j)\right)}{1 + \alpha}$$

So α is analogous to the precision (inverse variance) of a regular random variable. As α increases, the marginal probabilities $G(T_j)$ of the realizations G get increasingly close to those of the base distribution, $H(T_j)$. As α decreases, the distributions G generated by the DP become more wild, straying far from the base distribution.

Suppose that $\theta \sim G \sim \text{DP}(\alpha, H)$. That is, first we generate a random probability measure G from $\text{DP}(\alpha, H)$, then we draw a parameter θ from G. We can write this situation with the hierarchical model

$$\begin{aligned} G &\sim \text{DP}\,(\alpha, H) \\ \theta \mid G &\sim G \end{aligned} \tag{5.14}$$

Let's sample a few θ this way. Imagine that we see n_j of these values fall into $T_j \subseteq \Theta$. Then, using the Dirichlet distribution's conjugate prior property, we have

$$(G(T_1), ..., G(T_d)) \mid \theta_1, ..., \theta_n \sim \text{Dirichlet}\left(\alpha H(T_1) + n_1, ..., \alpha H(T_d) + n_d\right)$$

This is contingent on a particular partition T_1, ..., T_d, but we can remove this assumption in the following way. Let δ_{θ_i} be the identity probability measure for an observation θ_i, so that

$$\delta_{\theta_i}(T) = \begin{cases} 1 & \text{if } \theta_i \in T \\ 0 & \text{otherwise} \end{cases}$$

where $T \subseteq \Theta$.

Then we can rewrite the posterior distribution as

$$(G(T_1), ..., G(T_d)) \mid \theta_1, ..., \theta_n \sim \text{Dirichlet}\left(\alpha H(T_1) + \sum_{i=1}^{n} \delta_{\theta_i}(T_1), ..., \alpha H(T_d) + \sum_{i=1}^{n} \delta_{\theta_i}(T_d)\right) \tag{5.15}$$

Now this is true no matter which partition of Θ we pick, for any number of partition components d. Since this equation satisfies the definition of a DP 5.13, the posterior distribution is *also* a DP:

$$G \mid \theta_1, ..., \theta_n \sim \text{DP}\left(\alpha + n,\ \alpha H + \sum_{i=1}^{n} \delta_{\theta_i}\right) \tag{5.16}$$

Examining 5.16 closely, we see that the posterior base distribution is a weighted average between the prior base distribution H and the sample distribution formed by $\frac{1}{n}\sum_{i=1}^{n} \delta_{\theta_i}$. As the number of observations increases, the balance tilts away from the prior and toward the sample distribution.

These results are dependent on a particular draw $G \sim \text{DP}(\alpha, H)$. Imagine that we've already drawn n parameters from G, which we'll label as θ_1, ..., θ_n. If we *know* that we're dealing with a particular realization G, then the distribution of any future θ_{n+1} doesn't depend on what we've already seen. We still have

$$\theta_{n+1} \mid G, \theta_1, ..., \theta_n \sim G$$

As a result, for any $T \in \Theta$,

$$p\left(\theta_{n+1} \in T \mid G, \theta_1, ..., \theta_n\right) = G(T)$$

Marginalizing out G, we can take (loosely speaking)

$$\begin{aligned} p\left(\theta_{n+1} \in A \mid \theta_1, ..., \theta_n\right) &= \sum_G p\left(\theta_{n+1} \in A \mid G, \theta_1, ..., \theta_n\right) p(G \mid \theta_1, ..., \theta_N) \\ &= \sum_G G(A) p(G \mid \theta_1, ..., \theta_N) \\ &= \mathbb{E}\left[G(A) \mid \theta_1, ..., \theta_n\right] \end{aligned}$$

But due to 5.15, this is

$$\theta_{n+1} \,\Big|\, \theta_1, ..., \theta_n \sim \frac{\alpha H + \sum_{i=1}^{n} \delta_{\theta_i}}{\alpha + n} \tag{5.17}$$

Equation 5.17, first derived by Blackwell and MacQueen [87], is the central equation that enables inference for Dirichlet process priors. The predictive distribution for a new observation is given by the posterior base distribution. In other words, given some prior observations, the expectation of any future observation is distributed according to 5.17. Vitally, this distribution need not be discrete.

This observation is sometimes known as the *Blackwell-MacQueen urn scheme*, in reference to the following analogy. Suppose that each element in Θ is a unique color. Each draw θ will represent a ball of a particular color, which we will drop into an urn. Initially, there are no balls in the urn. For the first ball, we select a random color θ from the repertoire Θ with probability $H(\theta)$, and then drop it into the urn. For the second ball, we either select a new color from Θ with probability $\alpha H(\theta)/(\alpha + 1)$, or we paint it the color of the first ball and drop it into the urn. For the $(n+1)^{\text{th}}$ ball, we select a new color from Θ with probability $\alpha H(\theta)/(\alpha + n)$, or we randomly draw a ball from the urn, paint our new ball to match it, and drop them both back in. The probability to draw a ball with color j from the urn is proportional to n_j, the number of balls that already have that color in the urn. As n grows, the chance to select a new ball becomes exceedingly unlikely, and all of the probability mass for future observations becomes concentrated among the existing colors. The probability measure produced by this scheme is the realization of a Dirichlet process, $G \sim \text{DP}(\alpha, H)$.

A somewhat unexpected consequence is that while the base distribution H can be continuous, the realizations G are almost surely discrete. Only a finite number of different colors are represented in the urn. In other words, all of the probability mass is concentrated onto a countable number of points in Θ. (A proof can be found in [86].) Of course, exactly which colors are represented is different for each G. But by marginalizing over the individual G as in 5.17, we sidestep this problem and obtain continuous posterior estimates over all of the colors in the repertoire Θ.

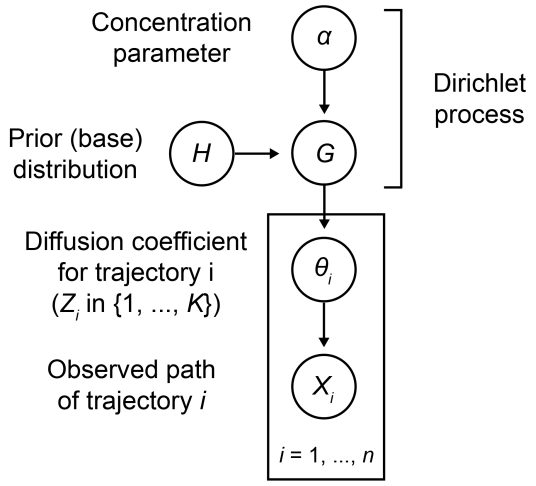

Figure 5.9: Graphical model for Dirichlet process mixture models. The discrete distribution G is generated from the base distribution H via a Dirichlet process. The goal of inference is to marginalize over G to get the posterior distribution of θ_i.

In practice, we usually cannot observe θ directly. Instead we observe various experimental consequences of it. A common example in this thesis is when θ represents a diffusion coefficient. A microscope doesn't hand us the diffusion coefficient on a platter, but instead presents us with some trajectories that are generated from it. If these trajectories are denoted by the random variable X, then the model is

$$\begin{aligned} G &\sim \mathrm{DP}(\alpha, H) \\ \theta \mid G &\sim G \\ X \mid \theta &\sim f_{X|\theta} \end{aligned} \tag{5.18}$$

where $f_{X|\theta}$ is the model likelihood, the various forms of which were investigated in detail in section 3.1. This scheme is illustrated in Fig. 5.9. For normal diffusion, $f_{X|\theta}$ is the probability to create a particular trajectory with diffusion coefficient θ. We'll refer to 5.18 as the *Dirichlet process mixture model* (DPMM).

Relation to finite-state mixtures

5.18 is actually simpler than the finite-state mixture model 5.12. We no longer have separate distributions over the state occupations and the state parameters, but they are joined into a single density function over all possible state parameters.

We now have the machinery to show how the DP mixture model emerges from a finite-state mixture as the number of states is allowed to become arbitrarily large [88]. Start with the finite-state mixture model 5.12. Given some sequence of

observations Z_1, ..., Z_n, we'd like the posterior distribution for a future observation Z_{n+1}. Using the multiplication law of probability,

$$p\left(Z_{n+1} \mid Z_1, ..., Z_n\right) = \frac{p\left(Z_1, ..., Z_n, Z_{n+1}\right)}{p\left(Z_1, ..., Z_n\right)}$$

Since each Z_i is just a categorical random variable with probabilities $\boldsymbol{\tau}$,

$$p\left(Z_1, ..., Z_n \mid \boldsymbol{\tau}\right) = \tau_{Z_1} \cdots \tau_{Z_n}$$

Marginalizing out $\boldsymbol{\tau}$, we have

$$\begin{aligned} p\left(Z_1, ..., Z_n\right) &= \int \left(\tau_{Z_1} \cdots \tau_{Z_n}\right) p\left(\boldsymbol{\tau}\right) \mathrm{d}\boldsymbol{\tau} \\ &= \frac{1}{\mathbf{B}\left(\frac{\alpha}{K}, ..., \frac{\alpha}{K}\right)} \int_0^1 \tau_{Z_1} \cdots \tau_{Z_n} \tau_1^{\frac{\alpha}{K}-1} \cdots \tau_K^{\frac{\alpha}{K}-1} \mathrm{d}\boldsymbol{\tau} \\ &= \Gamma(\alpha)\Gamma\left(\alpha/K\right)^{-K} \int_0^1 \tau_1^{\frac{\alpha}{K}+n_1-1} \cdots \tau_K^{\frac{\alpha}{K}+n_K-1} \mathrm{d}\boldsymbol{\tau} \end{aligned}$$

where n_j is the number of observations in Z_1, ..., Z_n that fall into category j. Using a property of the multivariate beta function, this is

$$p\left(Z_1, ..., Z_n\right) = \frac{\Gamma\left(\alpha\right) \Gamma\left(\frac{\alpha}{K} + n_1\right) \cdots \Gamma\left(\frac{\alpha}{K} + n_K\right)}{\Gamma\left(\frac{\alpha}{K}\right)^K \Gamma\left(\alpha + n\right)}$$

Then our conditional probability for a new observation is

$$p\left(Z_{n+1} = j \mid Z_1, ..., Z_n\right) = \frac{n_j + \alpha/K}{\alpha + n}$$

Letting $K \to \infty$, we have

$$\begin{aligned} \lim_{K\to\infty} p\left(Z_{n+1} \mid Z_1, ..., Z_n\right) &= \frac{n_j}{\alpha + n} \\ \lim_{K\to\infty} p\left(Z_{n+1} \neq Z_j \text{ for } j = 1, ...n\right) &= \frac{\alpha}{\alpha + n} \end{aligned} \tag{5.19}$$

The second equation is the probability that the new observation doesn't fall into one of the categories represented by the other observations. Following these equations procedurally for $n = 1, 2, ...$ produces the Blackwell-MacQueen urn scheme. As a result, the limit of the discrete-state mixture model 5.12 as $K \to \infty$ is the DP mixture model 5.18.

Number of nonzero components

When we use infinite mixture models, do we get an infinite number of components in the output? To answer this, consider the Blackwell-MacQueen urn scheme as represented in 5.19. Suppose that k_n represents the number of colors already represented in the urn after the n^{th} draw. Let E_{n+1} be the event that we pick a new color on the n^{th} draw, so that

$$\Pr\left(E_{n+1}\right) = \frac{\alpha}{\alpha + n}$$

Then $k_n = \sum_{i=1}^{n} \mathbb{I}_{E_i}$, where $\mathbb{I}$ is the indicator function. Linearity of expectation then provides

$$\begin{aligned}
\mathbb{E}\left[k_n\right] &= \sum_{i=1}^{n} \mathbb{E}\left[\mathbb{I}_{E_i}\right] \\
&= \sum_{i=1}^{n} \frac{\alpha}{\alpha + i - 1} \\
&= \alpha \sum_{i=1}^{n} \frac{1}{\alpha + i - 1}
\end{aligned}$$

Since this series is bounded by the harmonic series $\sum_{i=1}^{n} \frac{1}{i}$, which is itself bounded by $\log n + 1$ as $n \to \infty$, we have

$$k_n \sim O\left(\alpha \log n\right)$$

This provides support for our earlier assertion that the number of components in this kind of mixture becomes independent of K at large K. It also demonstrates the dependence of these mixtures on the concentration parameter. When α is low, we tend to find simple models consisting of a few components. When α is large, we find more complex models with many occupied components. The value of α is more significant than the amount of data for the resulting models, especially at large numbers of trajectories n.

5.5.2 Inference methods

The ability of DPs to model continuous posterior distributions is both the motivation for using them and a source of algorithmic challenges. These challenges were reviewed in detail by Neal [88], who also provided the basis for the strategies discussed here.

The Gibbs sampling schemes explored for arrayed state samplers clearly won't work for DPMMs because they would require that we evaluate the likelihood for each of an infinite number of states. A hint at the solution is provided by 5.17. Notice that the posterior distribution of θ_{n+1} given a current sample for $\theta_1, ..., \theta_n$ only requires that we consider the point densities produced by $\theta_1, ..., \theta_n$ along with the prior distribution. To see why this is useful, use θ_i to denote the parameters that produced each trajectory X_i. For each trajectory, we have one θ_i. Because the sequence of observations for a DPMM is exchangeable - that is, the probabilities remain the same no matter how we order the observations - we can always use 5.17 for a given trajectory i, imagining that all of the other trajectories came before it in the sequence:

$$\theta_i \;\Big|\; \{\text{all } \theta_j \text{ for which } j \neq i\} \sim \frac{\alpha H + \sum_j \delta_{\theta_j}}{\alpha + n - 1}$$

For convenience, let $\boldsymbol{\theta}_{-i}$ be the set of all θ_j for which $j \neq i$. Then, using Bayes' theorem,

$$\begin{aligned} p(\theta_i \mid X_i, \boldsymbol{\theta}_{-i}) &= \frac{p(X_i \mid \theta_i, \boldsymbol{\theta}_{-i})\, p(\theta_i \mid \boldsymbol{\theta}_{-i})}{p(X_i \mid \boldsymbol{\theta}_{-i})} \\ &= \frac{p(X_i \mid \theta_i)\, p(\theta_i \mid \boldsymbol{\theta}_{-i})}{p(X_i \mid \boldsymbol{\theta}_{-i})} \end{aligned}$$

The last part is due to the fact that, given known θ_i, X_i does not depend on any other $\theta_{j \neq i}$. Using 5.17 and dropping the evidence term, this is

$$p(\theta_i \mid X_i, \boldsymbol{\theta}_{-i}) \propto p(X_i \mid \theta_i) \left(\frac{\alpha H(\theta_i) + \sum_{j \neq i} \delta_{\theta_j}(\theta_i)}{n + \alpha - 1} \right) \tag{5.20}$$

The first term on the right side is just the model likelihood $f_{X|\theta}(X_i \mid \theta_i)$, while the second term is analogous to the role that the state occupancies $\boldsymbol{\tau}$ play in finite-state mixture models. Crucially, this form is amenable to Gibbs sampling: we can sequentially sample each θ_i while holding the other $\theta_{j \neq i}$ constant according to 5.20. The sequence of samples thus produced are samples from the posterior distribution $\theta \mid X_1, ..., X_n$.

Algorithmically, this scheme is:

1. Start with some set of parameter assignments $\theta_i^{(0)}$ for each trajectory X_i.
2. For each iteration $t = 1, ...$:
    (a) For each trajectory $i = 1, ..., n$

i. With probability $\alpha/(\alpha + n - 1)$, draw a new $\theta_i^{(t)}$ from Θ. The probability for each possibility θ is proportional to

$$H(\theta) f_{X|\theta}(X_i|\theta)$$

ii. With probability $(n-1)/(\alpha + n - 1)$, draw $\theta_i^{(t)}$ from the set of θ already represented in $\boldsymbol{\theta}_{-i}^{(t-1)}$. The probability of each possibility θ is proportional to

$$n_\theta f_{X|\theta}(X_i \mid \theta)$$

where n_θ is the number of observations already associated with θ.

3. Return the set of all $\boldsymbol{\theta}^{(t)}$, each of which is a sample of the posterior density over Θ.

This algorithm was used, for instance, by [89] and [90]. Although it works, the only way for us to generate new values of θ is by drawing from $H(\theta) f_{X|\theta}(X_i|\theta)$. If there are many observations associated with a high likelihood state θ', then many iterations are required before θ' becomes depleted enough by chance to explore nearby states that may also have high likelihood. As a result, the scheme is extremely inefficient.

As an alternative, we can endow the sampler with additional degrees of freedom that enable it to explore the posterior more rapidly. To do this, suppose that $\boldsymbol{\theta} = \{\theta_1, ..., \theta_K\}$ is a set of currently "active" parameters. This is analogous to the support for a particular draw $G \sim \text{DP}(\alpha, H)$. Let $Z_i \in \{1, ..., K\}$ be the assignment of trajectory i to one of the parameters in $\boldsymbol{\theta}$, so that the parameter corresponding to trajectory i is θ_{Z_i}. The set of all assignments $\mathbf{Z} = (Z_1, ..., Z_n)$ is now an auxiliary random variable in our model. When marginalized over $\boldsymbol{\theta}$, the elements Z_i can assume any of an infinite number of states. But given a particular $\boldsymbol{\theta}$, K is finite. Thus it is actually possible to generate samples from the conditional distribution $\mathbf{Z} \mid \boldsymbol{\theta}$ with a computer. Our goal is to construct a Gibbs sampler based on this scheme, drawing from the posterior distribution $p(\mathbf{Z}, \boldsymbol{\theta}|\mathbb{X})$ by sequentially sampling from $p(\mathbf{Z}|\boldsymbol{\theta}, \mathbb{X})$ and $p(\boldsymbol{\theta}|\mathbf{Z}, \mathbb{X})$.

Using Bayes' theorem, the probability to assign trajectory i to some state z with parameter θ_z is

$$\begin{aligned} p(Z_i = z \mid X_i, \mathbf{Z}_{-i}, \boldsymbol{\theta}) &= \frac{p(X_i \mid \theta_z)\, p(Z_i = z \mid \mathbf{Z}_{-i})}{p(X_i \mid \mathbf{Z}_{-i}, \boldsymbol{\theta})} \\ &\propto \begin{cases} p(X_i \mid \theta_z) \left(\frac{n_z}{\alpha+n-1}\right) & \text{if } \theta_z \in \boldsymbol{\theta} \\ p(X_i \mid \theta_z) \left(\frac{\alpha H(\theta_z)}{\alpha+n-1}\right) & \text{if } \theta_z \notin \boldsymbol{\theta} \end{cases} \end{aligned} \tag{5.21}$$

where the last is a consequence of 5.19. This gives us a mechanism to update the state assignments $\mathbf{Z}$. For the conditional distribution of $\boldsymbol{\theta}$ given $\mathbf{Z}$, for each $\theta_j \in \boldsymbol{\theta}$ we can use Bayes' theorem to write

$$\begin{aligned} p\left(\theta_j \mid \mathbf{Z}, \mathbb{X}, \boldsymbol{\theta}_{-i}\right) &\propto p\left(\mathbf{Z}, \mathbb{X} \mid \theta_j\right) p\left(\theta_j\right) \\ &= H(\theta_j) \prod_{i=1}^{n} \mathbb{I}_{Z_i=j} f_{X|\theta}(X_i|\theta_j) \end{aligned} \tag{5.22}$$

In other words, the likelihood for a particular choice of θ_j is the product of the likelihoods for all of the trajectories currently assigned to state j. These two equations give us the raw substrate for a Gibbs sampler, which would operate according to the following scheme:

1. Start with some set of states $\boldsymbol{\theta}^{(0)}$ and some set of state assignments $\mathbf{Z}^{(0)}$ for each trajectory.
2. For each iteration $t = 1, 2, ...$
   (a) For each trajectory $i = 1, ..., n$, sample $Z_i^{(t)}$ according to 5.21. If we pick a state not already in $\boldsymbol{\theta}^{(t-1)}$, add it to $\boldsymbol{\theta}^{(t-1)}$.
   (b) For each state $\theta_j^{(t-1)} \in \boldsymbol{\theta}^{(t-1)}$, sample a new $\theta_j^{(t)}$ according to 5.22.
3. Return the set of samples $\boldsymbol{\theta}^{(t)}$ along with their associated point densities $\mathbf{n}^{(t)}$, where $n_j^{(t)}$ is the number of observations corresponding to $\theta_j^{(t)}$.

By allowing the state parameters $\boldsymbol{\theta}$ to "move", exploration of the posterior with this scheme is much more rapid than the previous scheme. An important wrinkle remains: how do we sample θ_j according to 5.22, or in 5.21 when θ_z is not already in the existing set $\boldsymbol{\theta}$? $H(\theta)$ is a continuous distribution. When it is conjugate to the likelihood $p\left(X_i \mid \theta\right)$, we can just use the analytical posterior distribution in 5.22. In most cases, however, we won't choose a conjugate $H(\theta)$ - in fact, we'd like to use uniform priors to represent our complete lack of knowledge about the state parameters.

One approach in the case of 5.22 is to numerically integrate $H(\theta_j) \prod_{i=1}^{n} \mathbb{I}_{Z_i=j} f_{X|\theta}(X_i|\theta_j)$, which was pursued by numerous authors in the 90s. Neal [88] proposed a simpler approach: use a Metropolis-Hastings update step when selecting the new θ_j. While this increases the autocorrelation of the sampler's state sequence - that is, it often chooses $\theta_j^{(t)}$ that are close to the old values $\theta_j^{(t-1)}$ - it is far less costly in most cases than approaches based on numerical integration.

Of course, we can't use Metropolis-Hastings when drawing $\theta \sim H$ in 5.21. Instead, Neals implemented the following scheme: choose some random trial values $\theta_1, ..., \theta_{m_0}$ from Θ. Then set θ_z to one of these values with probability proportional to $H(\theta) f_{X|\theta}(X_i|\theta)$.

Algorithm 5.4 illustrates this scheme, which is equivalent to Algorithm 8 from Neal [88]. The end result is a sequence of diffusion coefficients $\boldsymbol{\theta}^{(t)}$ at each iteration t, along with $\mathbf{n}^{(t)}$, the number of trajectories assigned to each diffusion coefficient. The posterior distribution of θ can be then be approximated by taking a histogram over these samples. That is,

$$p\left(\theta \in [\theta_k, \theta_{k+1}] \mid \mathbb{X}\right) = \sum_t \sum_{j=1}^{K_t} n_j^{(t)} \mathbb{I}_{\theta_j^{(t)} \in [\theta_k, \theta_{k+1}]}$$

One can also use these samples to approximate the posterior by a kernel density estimate, although we dislike this approach because it requires that we choose a kernel.

Algorithm 5.4: General sampler for a Dirichlet process mixture model

Parameters: $\mathbb{X}$, an observed set of n trajectories; Θ, the set of permissible state parameters; H, the base distribution (prior) over Θ; α, the concentration parameter; $f_{X|\theta}(x|\theta)$, the trajectory likelihood function, where $\theta \in \Theta$; m_0, the number of trials to use when drawing from the prior, and g, a Metropolis-Hastings proposal distribution.

Algorithm:

1. Draw a set of random parameters $\boldsymbol{\theta}^{(0)} = (\theta_1, ...\theta_{m_0})$ with each element distributed according to $\theta \sim H(\theta) \prod_{i=1}^{n} f_{X|\theta}(X_i|\theta)$.

2. Assign each trajectory i to one of the elements of $\boldsymbol{\theta}^{(0)}$ with probability proportional to $H(\theta) f_{X|\theta}(X_i|\theta)$. Let this assignment be $Z_i^{(0)}$.

3. At each iteration $t = 1, 2, ...$

   (a) For each trajectory $i = 1, ..., n$, do one of the following:

   - With probability $n/(\alpha + n - 1)$, set $Z_i^{(t)}$ to an existing state in $\boldsymbol{\theta}^{(t-1)}$. The probability to select a particular state j is proportional to $n_j f_{X|\theta}(X_i|\theta_j)$ where n_j is the number of other trajectories assigned to θ_j.
   - Otherwise choose a new state for $Z_i^{(t)}$. Pick m_0 random parameter values from Θ. Among these, accept a particular value θ' with probability proportional to $H(\theta') f_{X|\theta}(X_i|\theta')$. Add this new state to $\boldsymbol{\theta}^{(t-1)}$.

   (b) For each $\theta_j \in \boldsymbol{\theta}^{(t-1)}$, if there are no observations currently assigned to state j, drop it. Otherwise add it to $\boldsymbol{\theta}^{(t)}$ and update it as follows:

   i. Propose a new $\theta_j' \sim g(\theta_j' \mid \theta_j^{(t-1)})$ and evaluate the likelihood ratio

$$r = \frac{\prod_{i=1}^{n} \mathbb{I}_{Z_i=j} f_{X|\theta}(X_i|\theta_j')}{\prod_{i=1}^{n} \mathbb{I}_{Z_i=j} f_{X|\theta}(X_i|\theta_j^{(t-1)})} \frac{g\left(\theta_j^{(t-1)} \mid \theta_j'\right)}{g\left(\theta_j' \mid \theta_j^{(t-1)}\right)}$$

   ii. Draw $u \sim \text{Uniform}(0, 1)$. If $r > u$, set $\theta_j^{(t)} = \theta_j'$. Otherwise set $\theta_j^{(t)} = \theta_j^{(t-1)}$. In either case, add $\theta_j^{(t)}$ to $\boldsymbol{\theta}^{(t)}$.

Return: The set of $\boldsymbol{\theta}^{(t)}$ and $\mathbf{n}^{(t)}$, where $n_j^{(t)}$ is the number of trajectories that were assigned to $\theta_j^{(t)}$ at iteration t.

5.5.3 Infinite mixture of regular Brownian motions

It is straightforward to apply Algorithm 5.4 to the case of mixtures of regular Brownian motions. Because it's useful as a example, we outline the result of this here, then show a few biological applications in the next section.

In the case of regular Brownian motion, we will deal with $\omega = \log(4D\Delta t + \sigma_{\text{loc}}^2)$ rather than the diffusion coefficient D directly. It is more useful to work with distributions defined on ω since the error associated with the estimate of D broadens rapidly with increasing D. Recalling the CRLB 3.19, the inherent variance associated with any estimate of D from a single trajectory goes as D^2. As a result, distributions over the diffusion coefficient defined on a linear scale tend to be highly nonintuitive for humans, since at higher D the probability density for any single D is dispersed over a large plot area. Working with a logarithmic scale mitigates this effect. An important note of caution is that it is not necessarily trivial to convert between a distribution defined in linear space and a distribution defined in log space, so the data should generally be presented according to the type of space in which the algorithm was actually run.

We will generally assume that the localization error σ_{loc}^2 is a constant here. Removing this assumption is a subject for future work.

For the parameter space Θ, we will choose all values of ω that correspond to a diffusion coefficient in the range D_{min} and D_{max}, which in this thesis we always take to be

$$D_{\text{min}} = 10^{-2}\ \mu\text{m}^2\ \text{s}^{-1}$$
$$D_{\text{max}} = 10^{2}\ \mu\text{m}^2\ \text{s}^{-1}$$

We choose this D_{max} because no biological proteins to date have been observed with diffusion coefficients higher than 100 $\mu\text{m}^2\ \text{s}^{-1}$. Our choice of D_{min} is motivated in the following way. Imagine we have a completely immobile object that presents itself to the microscope with localization error σ_{loc}^2. In section 3.2, we saw that the *apparent diffusion coefficient* of this object due to localization error alone is $\sigma_{\text{loc}}^2/\Delta t$, where Δt is the frame interval. For typical values of σ_{loc}^2 and Δt, this is usually between 0.02 and 0.2 $\mu\text{m}^2\ \text{s}^{-1}$. While it is still possible to infer diffusion coefficients below this value, one requires high statistics to do so. Accordingly, we set $D_{\text{min}} = 10^{-2}\ \mu\text{m}^2\ \text{s}^{-1}$.

If we have a trajectory i in m dimensions such that the sum of its radial squared jumps is S_i and there are L_i jumps total, then we have the RBM model likelihood

(equation 3.17)

$$f_{S|D}(S_i \mid D) = \frac{S_i^{\frac{mL_i}{2}-1} e^{-S_i/4(D\Delta t+\sigma_{\text{loc}}^2)}}{\Gamma\left(\frac{mL_i}{2}\right)\left(4(D\Delta t+\sigma_{\text{loc}}^2)\right)^{mL_i/2}}$$

This corresponds to the log likelihood

$$\log f_{S|\omega}(S_i \mid \omega) \propto -S_i e^{-\omega} - \frac{mL_i}{2}\omega \tag{5.23}$$

This proportionality can be taken as equality for our purposes, since the other terms are constant for a given trajectory. This means they are removed during any normalization over different states.

Since we have hard upper and lower limits for ω, we'll choose the Metropolis-Hastings proposal distribution

$$g(\omega' \mid \omega) = \begin{cases} \frac{1}{A\sqrt{2\pi\nu^2}} \exp\left(-\frac{(\omega'-\omega)^2}{2\nu^2}\right) & \text{if } \omega' \in [\omega_{\min}, \omega_{\max}] \\ 0 & \text{otherwise} \end{cases} \tag{5.24}$$

where

$$A = \Phi\left(\frac{\omega_{\max}-\omega}{\nu}\right) - \Phi\left(\frac{\omega_{\min}-\omega}{\nu}\right)$$

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2}\,\mathrm{d}t$$

The factor A accounts for the missing jump density below $\omega_{\min}$ and above $\omega_{\max}$, and its inclusion is necessary for computing the correct Metropolis-Hastings acceptance probability.

Finally, for the prior distribution, two different choices are reasonable for a noninformative prior:

- *Uniform priors.* Choose the base distribution H such that

$$H(\omega) = \begin{cases} \frac{1}{\omega_{\max}-\omega_{\min}} & \text{if } \omega \in [\omega_{\min}, \omega_{\max}] \\ 0 & \text{otherwise} \end{cases}$$

  This means that, lacking any trajectories, we have no reason to prefer a given ω over another. Notice that this is not the same as a uniform prior over the diffusion coefficient.

- *Defocalization-conscious priors.* Choose the base distribution H such that

$$H(\omega) = \begin{cases} \frac{1}{\omega_{\max} - \omega_{\min}} \frac{1}{\eta_\omega} & \text{if } \omega \in [\omega_{\min}, \omega_{\max}] \\ 0 & \text{otherwise} \end{cases}$$

  where η_ω is the probability for a trajectory with log variance ω to defocalize over one frame interval. This reflects the fact that, under a truly uniform prior, some trajectories are more likely to be observed than others due to defocalization.

Here, we always make the first choice. This means that the result, although somewhat more biased toward the slower states, is less dependent on our choice of parameters.

Algorithm 5.5 outlines the Dirichlet process mixture model sampler for regular Brownian motion with upper and lower bounds on the diffusion coefficient.

Algorithm 5.5: Dirichlet process mixture model sampler for regular Brownian motion

Parameters: $\mathbb{X}$, an observed set of n trajectories; Δt, the frame interval; σ^2_{loc}, the localization error; $\omega_{\min} = \log(4(D_{\min}\Delta t + \sigma^2_{\text{loc}}))$ and $\omega_{\max} = \log(4(D_{\max}\Delta t + \sigma^2_{\text{loc}}))$, the minimum and maximum diffusion coefficient; α, the concentration parameter; m_0, the number of trials to use when drawing from the prior; ν^2, the variance for the Metropolis-Hastings proposal distribution 5.24.

Algorithm:

1. Draw a set of random parameters $\boldsymbol{\omega}^{(0)} = (\omega_1, ..., \omega_{m_0})$ from a uniform distribution on the interval $[\omega_{\min}, \omega_{\max}]$.
2. Assign each trajectory i to one of the elements in $\boldsymbol{\omega}^{(0)}$ with log probability given by 5.23. Let this assignment be $Z_i^{(0)}$.
3. At each iteration $t = 1, 2, ...$
   (a) For each trajectory $i = 1, 2, ...$, either set $Z_i^{(t)}$ to an existing state in $\boldsymbol{\omega}^{(t-1)}$ with probability $(n-1)/(\alpha + n - 1)$, or create a new state with probability $\alpha/(\alpha + n - 1)$.
      - If setting to an existing state, choose state $\omega_j \in \boldsymbol{\omega}^{(t-1)}$ with log probability proportional to $\log n_j + \log f_{S|\omega}\left(S_i \mid \omega_j\right)$, where n_j is the number of jumps already assigned to state j
      - If choosing a new state, pick m_0 values of ω from the interval $[\omega_{\min}, \omega_{\max}]$. Among these, accept a particular value ω' with log probability proportional to $\log f_{S|\omega}\left(S_i \mid \omega'\right)$. Add this new state to $\boldsymbol{\omega}^{(t-1)}$.

   (b) For each $\omega_j \in \boldsymbol{\omega}^{(t)}$, if there are no trajectories currently assigned to state j, drop it. Otherwise add it to $\boldsymbol{\omega}^{(t)}$ and update it as follows:
      i. Propose a new $\omega' \sim g(\omega' \mid \omega_j^{(t-1)})$ and evaluate the likelihood ratio

$$r = \frac{\prod_{i=1}^{n} \mathbb{I}_{Z_i=j} f_{S|\omega}(S_i|\omega')}{\prod_{i=1}^{n} \mathbb{I}_{Z_i=j} f_{S|\omega}(S_i|\omega_j^{(t-1)})} \frac{\Phi\left(\frac{\omega_{\max}-\omega_j^{(t-1)}}{\nu}\right) - \Phi\left(\frac{\omega_{\min}-\omega_j^{(t-1)}}{\nu}\right)}{\Phi\left(\frac{\omega_{\max}-\omega'}{\nu}\right) - \Phi\left(\frac{\omega_{\min}-\omega'}{\nu}\right)}$$

      ii. Draw $u \sim \text{Uniform}(0, 1)$. If $r > u$, set $\omega_j^{(t)} = \omega_j^{(t-1)}$. Otherwise set $\omega_j^{(t)} = \omega_j'$. In either case, add $\omega_j^{(t)}$ to $\boldsymbol{\omega}^{(t)}$.

Return: The set of $\boldsymbol{\omega}^{(t)}$ and $\mathbf{n}^{(t)}$, where $n_j^{(t)}$ is the total number of jumps assigned to $\theta_j^{(t)}$ at iteration t.

5.5.4 Note on DPMM model complexity

When analyzing data with Dirichlet process mixture models, we generally want to minimize model complexity while describing the data as accurately as possible. However, "model complexity" can have a slightly different meaning from the perspective of a Bayesian mixture model than for a human. For a Bayesian mixture model like eq. 5.12, model complexity means the *number of states with significant occupancy in the posterior model*. When two models - one with 10 states and one with 5 states - describe the data equally well, the one with 5 states is preferred.

This property carries over to Dirichlet process mixture models (eq. 5.18). However, because the state distribution is now continuous rather than discrete, the models with minimum complexity tend to be described a few disjoint peaks rather than a continuous smear. This may not be the desired behavior. For instance, when we *expect* smears of diffusion coefficients, or would rather not refine our estimate of the diffusion coefficient of a individual state unnecessarily. The worst that a method can do is to create peaks - which a human interprets as distinct states - where in reality there is a continuous smear of diffusion coefficients. This is analogous to the problem of cluster identification in unsupervised machine learning.

We find that unmodified Dirichlet process samplers often exhibit this issue on small datasets (<10000 trajectories) with few real peaks when the concentration parameter is too low. In short, the DPMMs appear to identify sporadic peaks in smears, creating an almost Gibbs phemonenon-like appearance. While the issue can always be overcome with additional data, users often will not have sufficient data or will not be sufficiently aware of the problem.

To correct this behavior, we can introduce a method that dampens the "peak detection" property of DPMMs. The resulting methods are more conservative when identifying diffusion coefficients that stand out among the rest. This can be viewed as essentially a safety feature - by damping the sensitivity of the DPMM, we can increase its reliability on small datasets.

The modification is the following. Change step 3(a) in Algorithm 5.4. If X_i is the i^{th} trajectory, then rather than selecting one of the existing states with probability proportional to

$$n_j f_{X|\theta}\left(X_i|\theta_j\right)$$

where n_j is the number of jumps already assigned to state j, instead select a state j with probability proportional to

$$\max\left(n_{\max}, n_j\right) f_{X|\theta}\left(X_i|\theta_j\right)$$

The rest of Algorithm 5.4 proceeds normally. Intuitively, this modification limits the extent to which trajectories influence each other's choice of state. While arbitrary, it has the convenient limiting cases:

- When $n_{\text{max}} \to 1$, the posterior distribution is just the likelihood function for the diffusion model. This treats all trajectories as independent.
- When $n_{\text{max}} \to \infty$, the posterior distribution is the true posterior distribution for a Dirichlet process mixture model of the type 5.18.
- For all n_{max} between 1 and ∞, the posterior distribution is intermediate between the likelihood function and the true posterior.

In experimental datasets, we find it useful to set $n_{\text{max}} = 100$. From a computational perspective, lower values of n_{max} will result in higher numbers of active states at each iteration, lowering the speed. However, this rarely limits the algorithm to taking more than a couple of minutes to run on a laptop.

5.5.5 Examples

In this section, we show some applications of the Dirichlet process mixture models to simulated datasets. We investigate the application of these models on real spaSPT datasets in the next chapter.

First, we applied the DPMM sampler (Algorithm 5.5) to simple mixtures of regular Brownian motions (Fig. 5.10). In these experiments, we simulated trajectories under realistic SPT constraints, including defocalization, localization error, and photobleaching (see caption for Fig. 5.10).

The method was able to recover the distribution of states, even in the presence of mixtures of six diffusing states. Comparison with another method to recover distributions of diffusion coefficients, the MSD histogram method (Fig. 5.11)) revealed the superior resolution of the DPMM sampler. However, if the states were simulated with diffusion coefficients that were very close (Fig. 5.12), the DPMM sampler tended to aggregate nearby states into single states. Although increased resolution can be achieved by setting n_{max} higher, we prefer to limit the resolution of the DPMM sampler in exchange for more regular behavior on complex mixtures.

Next, we applied the DPMM sampler to non-discrete distributions of diffusion coefficients (Fig. 5.13). In these simulations, trajectories are not drawn from a finite number of states, but from continuous distributions over the diffusion coefficient as indicated in the "generate density" subplots. The DPMM sampler (with $n_{\text{max}} = 100$) faithfully recovered the distributions of diffusion coefficients, even

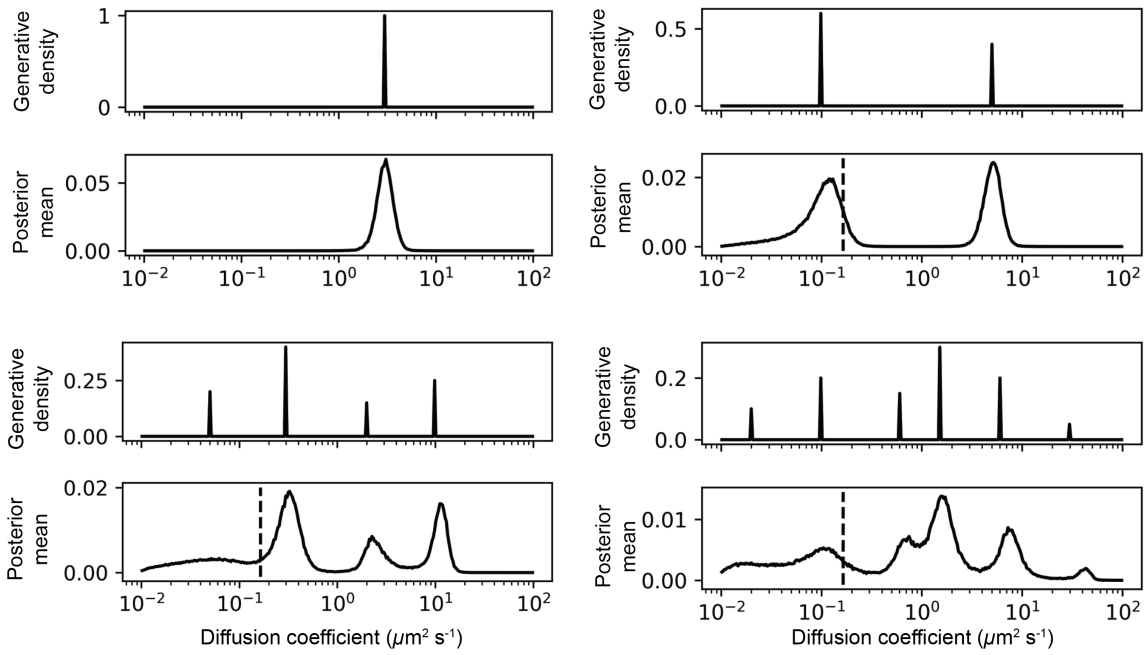

Figure 5.10: Application of the Dirichlet process mixture model sampler to simple mixtures of regular Brownian motions. In each panel, the upper plot indicates the simulated distribution of diffusion coefficients and the bottom is the posterior density estimated from the output of a Dirichlet process mixture model sampler (Algorithm 5.5). The following was simulated: trajectories corresponding to the generative density were photoactivated with uniform probability throughout a spherical nucleus (5 μm radius), subject to a 13.4 Hz bleaching rate. Localizations within a 700 nm focal depth bisecting the nucleus were recorded and tracked. The simulated frame interval was 7.48 ms, and each localization has associated with it 35 nm of 1D localization error (root variance). Under these conditions, the mean trajectory length is about 3-4 frames, and about 10000 trajectories were used for inference. Six instances of the DPMM sampler 5.5 were run for 1000 iterations with a 20 iteration burn-in period, $\alpha = 10$, and $n_{\text{max}} = 100$ (as described in the previous section). The samples produced by the DPMM were used to obtain a posterior mean estimate over the diffusion coefficient by discretizing the posterior density in log-spaced bins (with no kernel density estimation). The dotted line is the apparent diffusion coefficient of a completely immobile object due to localization error ($\sigma^2_{\text{loc}}/\Delta t$, where Δt is the frame interval).

when the underlying distribution was a log-uniform or log-triangular distribution. The selection of a low n_{max} value, and the use of >10000 trajectories, is important for the accuracy of the method when applied to these types of simulations.

Non-discrete distributions of diffusion coefficients also provide a good opportunity to demonstrate the role of the defocalization terms in the sampler (Fig. 5.14). These take the form of the η terms in Algorithm 5.5, which can be computed with

either 4.1 or 4.2 (depending on whether gaps are used during tracking). Without accounting for defocalization, the shallow depth of field of this simulated microscope setup results in a strong bias against fast-moving molecules, leading to overestimation of the occupation of slow diffusion coefficients.

Altogether, we find that the DPMM sampler performs at least as well as the arrayed state samplers for recovering complex distributions of the diffusion coefficient. Because the only parameters governing the DPMM are the concentration parameter α and the coupling term n_{max}, the DPMM sampler represents a more nonparametric approach to spaSPT analysis than the approaches outlined in the previous chapter.

Finally, we summarize four important aspects of the DPMM sampler for users:

- There is a "resolution limit" to the DPMM sampler. If two states have very similar diffusion coefficients, the DPMM sampler will tend to see a single

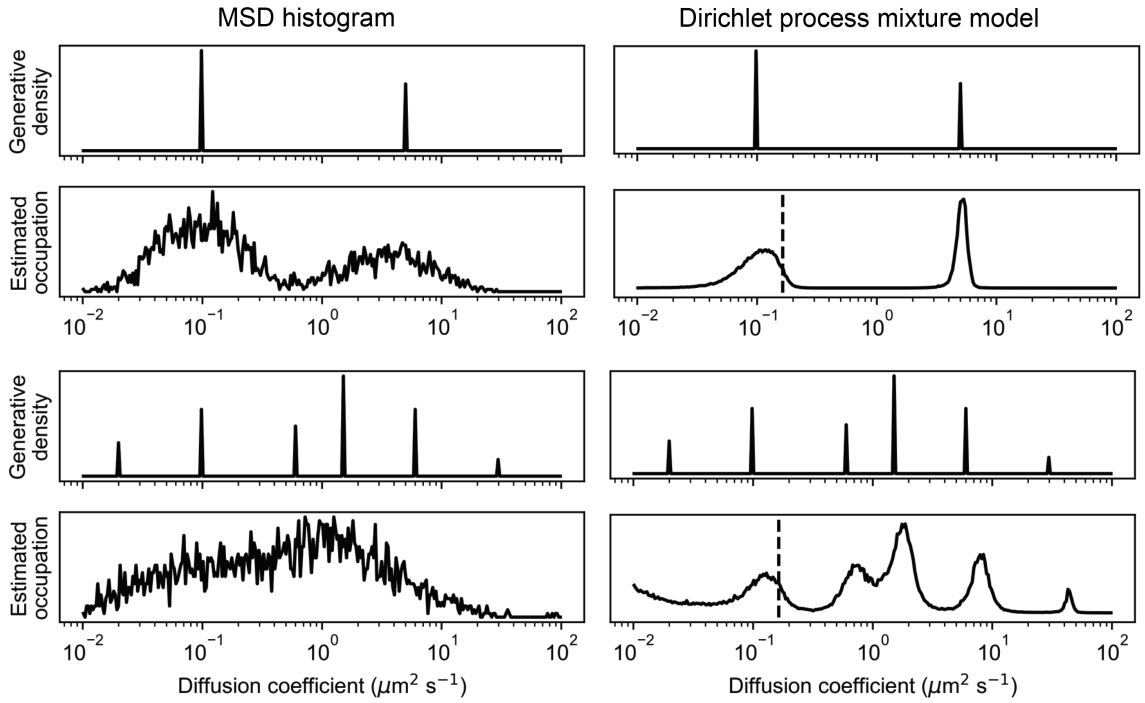

Figure 5.11: Comparison of Dirichlet process mixture models with the MSD histogram approach. Simulations and DPMM were performed as in Fig. 5.10. Trajectories were then either subjected to a run of Algorithm 5.5 as in Fig. 5.10, or were subjected to the "MSD histogram" approach to fitting. Briefly, for the latter approach, we removed all trajectories shorter than 4 jumps, computed the MSD, and fit to a linear model $\text{MSD}(t) = 4(\sigma_{\text{loc}}^2 + D\Delta t)$. The fits for D were then binned according to the same scheme as the DPMM posterior density and plotted.

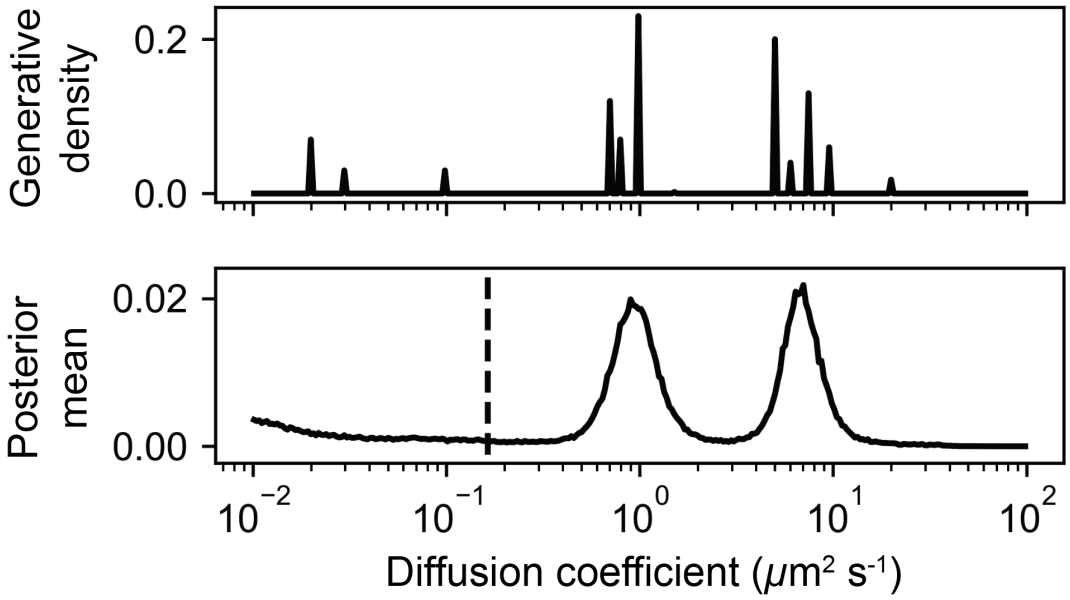

Figure 5.12: Illustration of the "aggregating" effect of Dirichlet process mixture models for nearby states. Simulations and DPMM inference were performed as in Fig. 5.10. A total of twelve distinct diffusive states were simulated. The upper subplot is the "ground truth" (simulated density), while the bottom is the posterior mean from a run of Algorithm 5.5. Notice that when there are multiple states that have similar diffusion coefficients, the algorithm tends to aggregate them into a single peak.

state.

- $n_{\max}$ can be used to tune the "confidence" of the DPMM sampler. Higher values will make the DPMM sampler more confident in calling peaks.
- Depth of field and localization error are important constants in the DPMM. They must be determined ahead of time.
- The DPMM sampler, like other methods, has difficulty inferring the distribution of diffusion coefficients below $\sigma_{\text{loc}}^2/\Delta t$ (where σ_{loc}^2 is the localization error and Δt is the frame interval).

5.6 Summary

We saw in the last chapter that, given the correct model for a set of spaSPT data, there are fairly robust frameworks for recovering the model parameters. Deter-

mining the correct model in the first place is more challenging. The most important questions when choosing a model for spaSPT data are

- How many states?
- What kind of motion? (regular Brownian, fractional Brownian, etc.)

Existing frameworks for interpreting spaSPT data, such as radial jump histogram fits or MSD histogram fitting, lack robust ways to distinguish between the viability of different models apart from examination of fit residuals or arbitrary penalties such as the AIC or BIC. Indeed, such methods tend to exploit all of the degrees of freedom with which we provide them. As a result, the inferred parameters tend to be sensitive to noise and vary strongly between experiments.

In this chapter, we explored some Bayesian alternatives to least squares-based methods. By construction, these alternatives have a natural tradeoff between model complexity and the model's likelihood given a dataset. The resulting models tend to be sparse. For instance, in the VB algorithm with 16 states, most of the states have zero occupation in the posterior model. Exactly how many states are "used" by the algorithm depends on whether the algorithm can find sufficient evidence for them in the data. As we have more data, the algorithm becomes more confident in using more states.

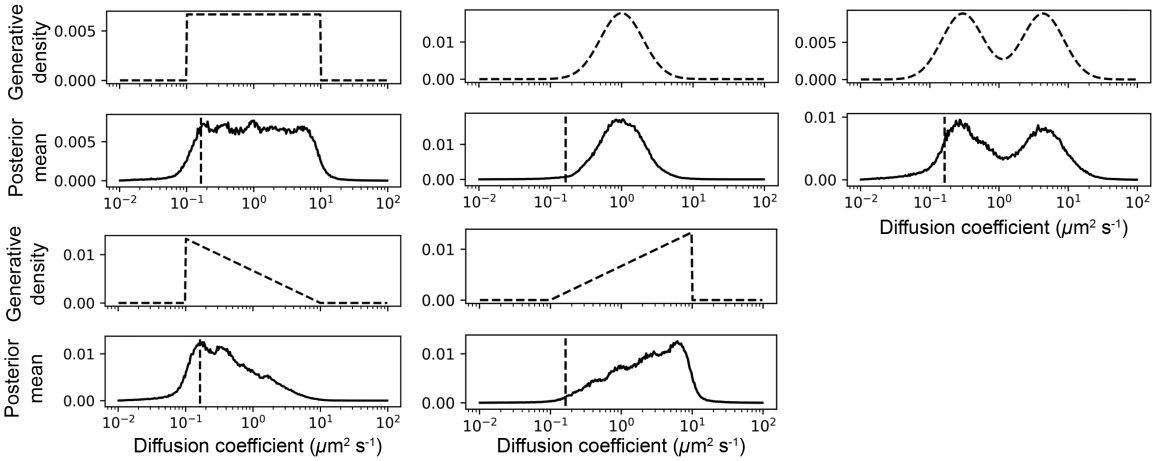

Figure 5.13: Application of Dirichlet process mixture model samplers to non-discrete distributions of diffusing states. Simulations and DPMM inference were performed as in Fig. 5.10, except instead of simulated a discrete number of diffusive states, the diffusion coefficients for each trajectory were drawn from the distribution indicated by the dotted line in the upper subplots. As in Fig. 5.10, about 10000 trajectories were used for inference.

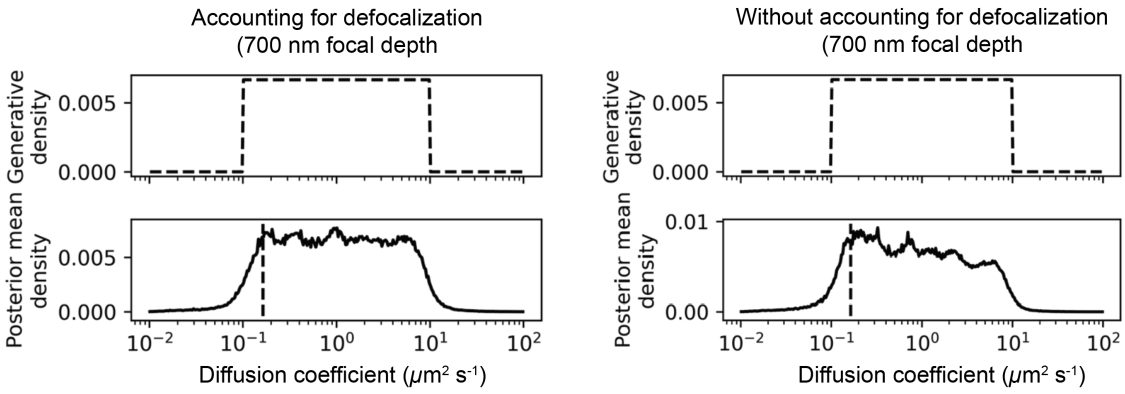

Figure 5.14: Demonstration of the role of the defocalization correction. Simulations and DPMM inference were performed as in Fig. 5.13, with and without the η factors to correct for defocalization. Notice that, similar to the observations of [59] and [60], without explicitly accounting for defocalization, we end up overestimating the fraction of slower-moving molecules.

This does not mean that these methods always return the true number of states. However, the accuracy with which they estimate the true model complexity increases as more trajectories are collected. In the case of Dirichlet process mixture models (DPMMs), probably at least 10000 trajectories are required for robust inference. This is on the order of 5 - 15 spaSPT experiments, depending on the length of acquisition and localization density.

While most of the methods in this chapter descend from the field of "nonparametric" Bayesian statistics, in reality they are not actually nonparametric and depend critically on the user choice of variables such as the concentration parameter α and the maximum occupation weight n_{max}. In all cases, the reliability of the methods with a given parameter selection should be determined by biological replicates.

We recommend the following when approaching a completely novel protein in a novel setting (such as an endogenously tagged cell line):

1. Evaluate the regular Brownian motion likelihood function for all of the individual tracking files in the dataset. This accomplishes two things: (1) allows the experimenter to assess file-to-file variability, and (2) allows the experimenter to determine whether there are multiple diffusing states present. Importantly, this method doesn't involve statistical inference and is simple to compute.

2. Evaluate the jump covariance matrix and angular distribution for the protein, as described in section 3.2. This enables the experimenter to determine whether memory effects are likely to play a role in the motion of their protein.

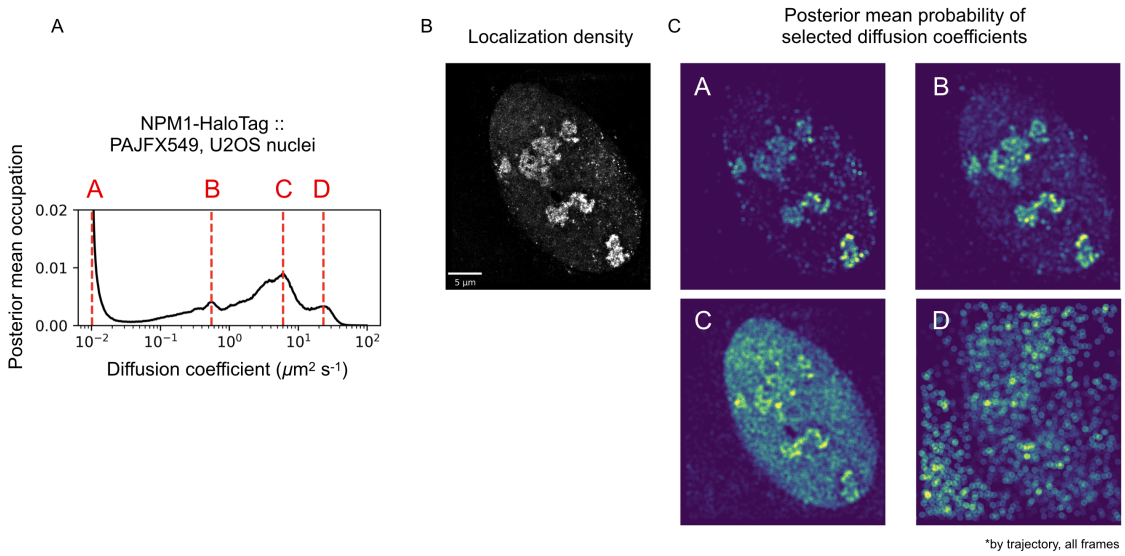

Figure 5.15: Assessing spatial bias in posterior model probabilities. Endogenously tagged nucleophosmin (NPM1)-HaloTag cells labeled with the photoactivatable PA-JFX549 dye were assayed by spaSPT with 7.48 ms frame intervals. (A) Dirichlet process mixture model posterior density for all trajectories from one movie, using a regular Brownian motion likelihood function. (B) Gaussian kernel density estimate for localization density, pooling all localizations from the same movie. (C) Posterior model likelihood function for each trajectory at the indicated diffusion coefficients, plotted as a function of space. Each subplot represents the likelihood function evaluated at the corresponding diffusion coefficient from (A). Localizations have been convolved with a circular kernel of radius 320 nm for visualization.

If so, fractional Brownian motion is a model that can capture these effects. In contrast, if they are negligible, then regular Brownian motion models are more useful.

3. If the motion is regular Brownian, then we recommend application of the variational Bayes algorithm 5.1, then application of the Dirichlet process mixture model 5.5.

4. If the motion has non-negligible memory effects, then we recommend application of the arrayed state sampler for fractional Brownian motion 5.3.

5. In both of the above cases, the posterior model should be compared with the raw likelihood function.

What should be clear from this list is that it is vital to compare multiple methods to analyze a dataset. Anomalies in the output of one method can potentially be windows into unexpected, novel behavior.

Chapter 6

Competition in type II nuclear receptors

6.1 Introduction

Type II nuclear receptors (T2NRs) are a family of ligand-activated transcription factors in vertebrates that includes retinoic acid receptor (RAR), vitamin D receptor (VDR), thyroid hormone receptor (TR), and others [91]. All T2NRs are believed to require heterodimerization with a common factor, the retinoid X receptor (RXR), to bind chromatin and regulate their target genes (Fig. 6.1A). Because the pool of RXR is shared among all T2NRs, competition between individual T2NRs has the potential to limit access to the chromatin-bound state (Fig. 6.1B,C).

Such competition, if it exists, would be important for two reasons. First, it would result in an interdependence between the activity of T2NRs. There is some suggestion that this may be important in development. For instance, the orphan T2NR Nr0b1 (homolog of the human DAX1 protein) has been implicated in early murine embryogenesis and stem cell pluripotency [92] [93] [94]. NR0B1 is a T2NR that lacks a DNA-binding domain but retains a ligand-binding domain capable of dimerization with other T2NRs, and was shown to inhibit retinoic acid- induced activation of RARA in its cloning paper [95]. A potential explanation for this result is that NR0B1 competes away RXR from the other T2NRs by heterodimerization, reducing the sensitivity of mESCs to differentiating agents such as retinoic acid. The idea that competition for RXR modulates endogenous T2NR-mediated gene regulation is also supported by the observed antagonism between the gene expression of combinations of T2NRs in luciferase assays [97] [96].

The second reason is that competition between T2NRs has been proposed to underlie the inactivation of RAR in acute promyelocytic leukemia (APL). APL is characterized by chromosomal rearrangements that join the 3' exons of the *RARA*

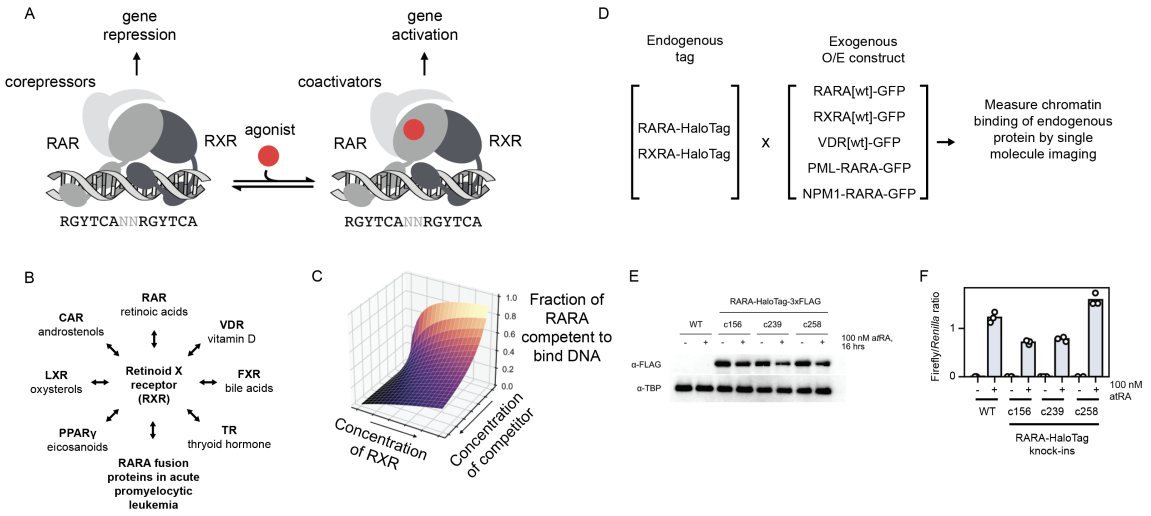

Figure 6.1: Schematic of the type II nuclear receptor network. (A) Cartoon of a type II nuclear receptor heterodimer composed of retinoic acid receptor (RAR) and retinoid X receptor (RXR). Agonists toggle the specificity of the RAR ligand-binding domain from corepressors to coactivators. Common physiological agonists include retinoic acids. (B) Simplified version of the type II nuclear receptor network. Double-headed arrows between two nodes indicate that that the nodes dimerize. RARA fusion proteins in acute promyelocytic leukemia do not exist naturally, but are produced by chromosomal rearrangement and can dimerize with RXR. (C) Hypothetical competitive equilibrium governing RARA chromatin binding. (D) Schematic of the experimental approach in this paper. RARA and RXRA are endogenously tagged, and then their expression and binding dynamics are assayed in the presence of various competitors. (E) Anti-FLAG Western blot to determine the presence of endogenously tagged RARA-HaloTag-3xFLAG in U2OS cells. "atRA" is all-trans retinoic acid, a RARA agonist that also causes RARA degradation. (F) Luciferase assays with a retinoic acid response element (RARE)-driven reporeter to assess the response of endogenously tagged RARA-HaloTag-3xFLAG cell lines to retinoic acid.

locus to 5′ exons from various other genes, producing fusion proteins [99]. Cells containing RARA fusion proteins are resistant to differentiation induced by physiological concentrations of the agonist all-trans retinoic acid (atRA), requiring either therapeutic concentrations of atRA or other drugs (such as arsenic trioxide). Competition for RXR would explain why the presence of RARA fusion proteins apparently inhibits activity by the remaining wildtype *RARA* allele. Indeed, a mutant of the PML-RARA fusion protein defective for RXR dimerization failed to trigger APL development in transgenic mice [100].

Nevertheless, there appear to be additional determinants of T2NR crosstalk in live cells beyond RXR heterodimerization. Studying competition between thyroid hor-

mone receptor α (TRA) and peroxisome proliferator-activated receptor γ (PPARG), Hunter et al. found that despite TRA and PPARG antagonism of each each other's DNA binding in vitro, they exhibited a cooperative effect on gene expression of PPARG target genes in human breast cancer cell lines [98]. These results highlight that the codependence of T2NRs in their native context can depart markedly from models constructed purely from in vitro experiments.

Here, we examine the interdependence of T2NR dynamics by endogenously tagging the RAR and RXR genes with HaloTag and tracking their movement in live cells using fluorescent single particle tracking (spaSPT). These results provide a complement to existing dynamics experiments in vitro and in vivo.

6.2 Results

6.2.1 Endogenous tagging of RARA-HaloTag in U2OS osteosarcoma cells

To assess whether T2NRs compete for a common pool of RXR, we endogenously tagged retinoic acid receptor alpha (RARA) with HaloTag-3xFLAG in U2OS osteosarcoma cells (Fig. 6.1). We took three homozygous clones (c156, c239, and c258) for subsequent assays. These clones have a similar expression level of RARA-HT, which is reduced by approximately 50% in the presence of all-trans retinoic acid (Fig. 6.1E). All clones responded to all-trans retinoic acid by upregulating expression of a retinoic acid response element (RARE)-driven luciferase reporter (Fig. 6.1F), albeit to an extent that depended on the clone in question. Using quantitative spinning disk confocal microscopy to quantify expression levels across several knock-in clones showed that the expression level of RARA-HT is similar between subclones in U2OS cells (Fig. 6.2D).

To further assess whether RARA was still functional when fused to HaloTag-3xFLAG, we performed luciferase assays with transfected transgenes containing either the wildtype RARA, RARA-HaloTag-3xFLAG, or RARA[C88G]-HaloTag-3xFLAG (Fig. 6.2B). C88G is a point mutation in the zinc fingers that abolishes DNA binding in vitro [102]. Transfection of either wildtype RARA or RARA-HaloTag-3xFLAG reduces expression of the RARE-driven luciferase reporter in response to atRA, perhaps due to a squelching effect. The reduction was the same for either wildtype RARA or RARA-HaloTag. In contrast, expression of the nonfunctional RARA[C88G]-HaloTag-3xFLAG reduced the expression by a larger extent.

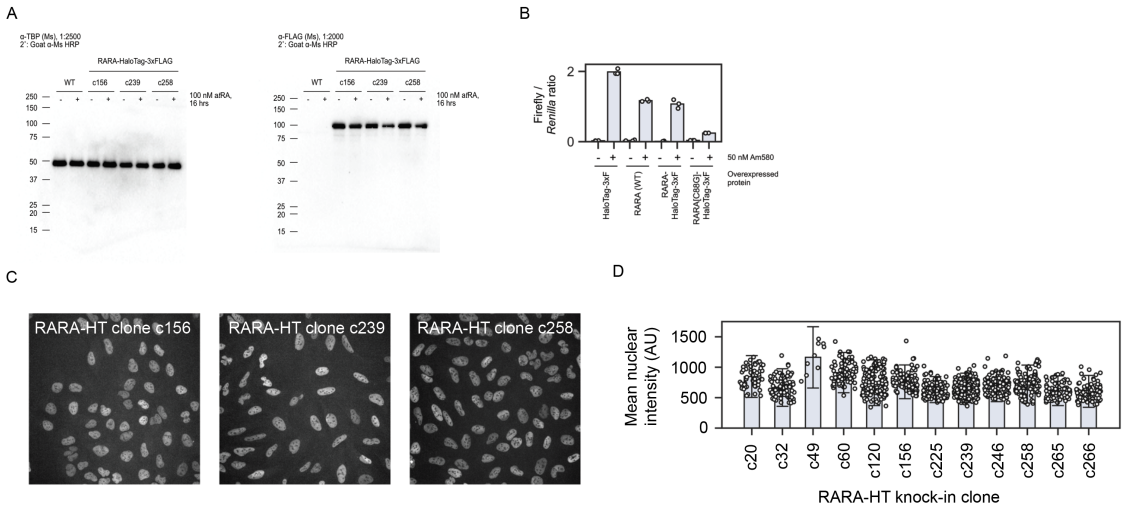

Figure 6.2: Supplementary plots for U2OS retinoic acid receptor alpha knock-in cell lines. (A) Full Western blots from 6.1(E). (B) Luciferase assays to assess the ability of transgene RARA to activate transcription in response to agonist. Am580 is a synthetic retinoid analog agonist. Each data point is a biological replicate. Transgenes have been expressed under a EF1a promoter on a PiggyBac vector. (C) Spinning disk confocal microscopy images of TMR-labeled endogenously tagged U2OS RARA-HT cell lines. Colors have been scaled identically for all subplots. (D) Quantification of spinning disk confocal microscopy images of TMR-labeled endogenously tagged U2OS RARA-HT cell lines. Each bar represents a separate clone, and each data point represents a separate nucleus for that clone.

6.2.2 Heterogeneity of diffusive states for RARA-HaloTag

First, we established a baseline expectation for the diffusive behavior of RARA-HaloTag in unperturbed U2OS nuclei. We performed spaSPT experiments on RARA-HaloTag labeled with photoactivatable PA-JFX549 [15] at 7.48 ms frame intervals with 1.5 ms pulse widths (Fig. 6.3). Under these conditions, the mean trajectory length is 3-4 frames and the focal depth is $\sim$ 700 nm.

To assess heterogeneity between individual nuclei in the same dataset, we used the aggregate likelihood method with regular Brownian motion (Fig. 6.3D). Compared to HaloTag-NLS or HaloTag alone, RARA-HaloTag-3xFLAG exhibits a broad range of diffusion coefficients and a substantial immobile fraction present across most nuclei. In contrast, slower-diffusing states were inconsistently observed between nuclei in the HaloTag and HaloTag-NLS datasets. Similar to other reports [19], we observed that the addition of an NLS to HaloTag reduces its diffusion coefficient by 2-3 times.

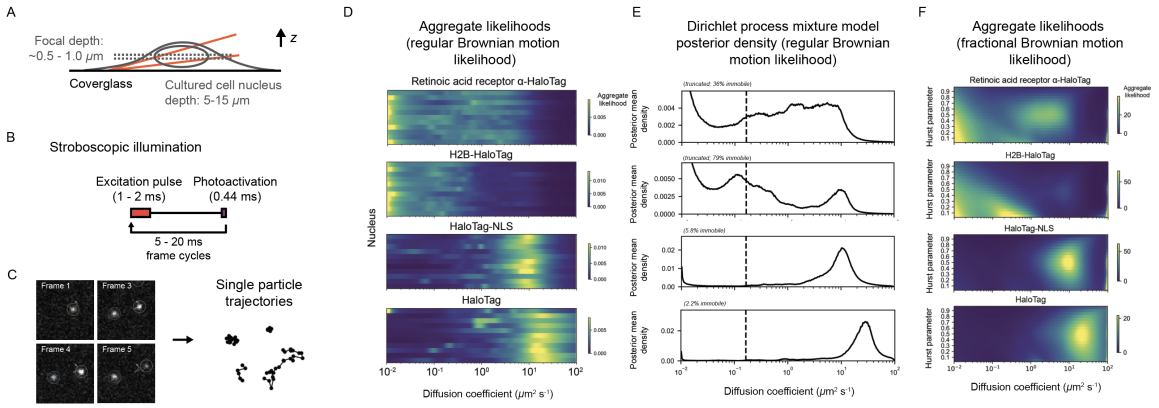

Figure 6.3: Assessing the native dynamics of RARA-HaloTag in U2OS nuclei. (A) Cartoon of the imaging geometry for an spaSPT experiment with HiLo geometry. (B) Schematic for the illumination sequence. All excitation is concentrated in a period of 1-2 ms at the beginning of each frame interval to minimize motion blur. Photoactivation of PA-Janelia Fluor [15] dyes is relegated to the frame transition times. (C) Sample images of individual spots observed with the tracking algorithm. (D) Aggregate regular Brownian motion likelihoods for endogenously tagged RARA-HaloTag c258 and three control proteins (histone H2B, HaloTag-NLS, and HaloTag) evaluated on a log-spaced grid of diffusion coefficients. Each row represents an spaSPT experiment on an individual nucleus. Colors have been scaled individually for each subplot. (E) Output of a Dirichlet process mixture model run on the same datasets. Trajectories across all nuclei have been aggregated for the Dirichlet process mixture model runs. The posterior mean density has been estimated by binning into log-spaced diffusion coefficient bins, without kernel density estimation. The dotted line represents the apparent diffusion object of a completely immobile object due to localization error at this frame interval. (F) Aggregate fractional Brownian motion likelihoods for the same conditions. In this case, the localization error was held constant at 30 nm (one dimensional root variance). Trajectories from all files in each condition have been aggregated for each condition. The color map has been scaled independently for each subplot.

Surprisingly, while both RARA-HaloTag and histone H2B-HaloTag both have immobile fractions, they differ markedly in the distribution of mobile diffusion coefficients. In particular, diffusion coefficients in the range 1.0 to 5.0 μm^2 s^{-1} are rare for H2B-HaloTag, although there is a low-occupancy fast diffusing state at $\sim$10 μm^2 s^{-1}.

To more precisely quantify the fraction of molecules in different diffusing states, we pooled trajectories across the nuclei in each of these datasets and analyzed this pool with a Dirichlet process mixture model. In this case, we used a regular Brownian motion likelihood function and a branch probability of 0.1 (Fig. 6.3E). This analysis revealed small but consistent immobile fractions for HaloTag and

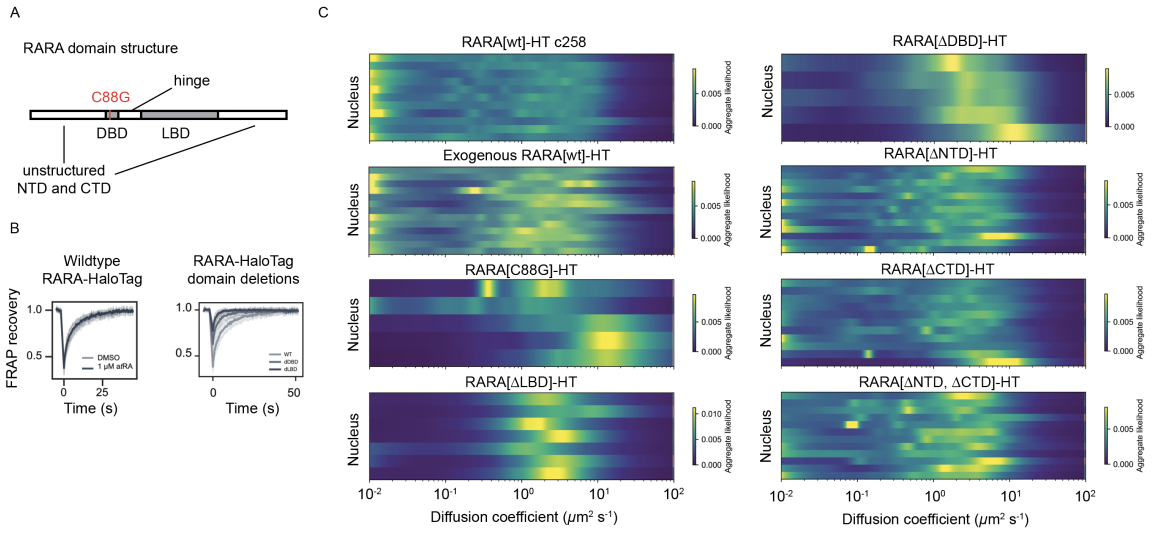

Figure 6.4: Effect of mutations and domain deletions on RARA-HT dynamics. (A) Schematic of RARA domain structure. DBD is the DNA-binding domain and LBD is the ligand-binding domain, which is also responsible for dimerization with RXR. C88G is a point mutation that abolishes DNA binding in vitro. (B) Fluorescence recovery after photobleaching for transfected wildtype RARA-HaloTag or RARA bearing domain deletions. (C) Aggregate regular Brownian motion likelihoods for either endogenously tagged RARA-HT or exogenously expressed RARA-HT bearing domain deletions.

HaloTag-NLS, which may be stuck dye molecules. In addition, we observed that most histone H2B-HaloTag molecules are immobile and about 30-35% of RARA-HaloTag molecules are immobile. Consistent with the aggregate likelihood analysis, RARA-HaloTag exhibits a "smear" of diffusion coefficients in the range 0.1 to 10.0 μm^2 s^{-1}, while H2B-HaloTag shows two major modes centered at approximately 0.2 and 10.0 μm^2 s^{-1}. Interestingly, only the upper state appears to have a Hurst parameter around 0.5, consistent with regular Brownian motion (Fig. 6.3F).

To gain additional insight into the origin of the state landscape for RARA-HaloTag, we created transgenes with RARA bearing either point mutations or domain deletions and assayed dynamics with spaSPT (Fig. 6.4). Since these constructs were expressed exogenously, we compared the dynamics of both exogenously and endogenously expressed RARA-HaloTag against these domain deletions.

Using the aggregate likelihood method, we observed a generally similar profile of diffusion coefficients for exogenously expressed RARA-HaloTag as endogenous RARA-HaloTag, albeit with a bias toward faster-moving states. Deletion of either the DNA-binding domain (DBD) or ligand-binding domain (LBD) abolished the immobile fraction of RARA-HaloTag, as did mutation of a key residue involved in

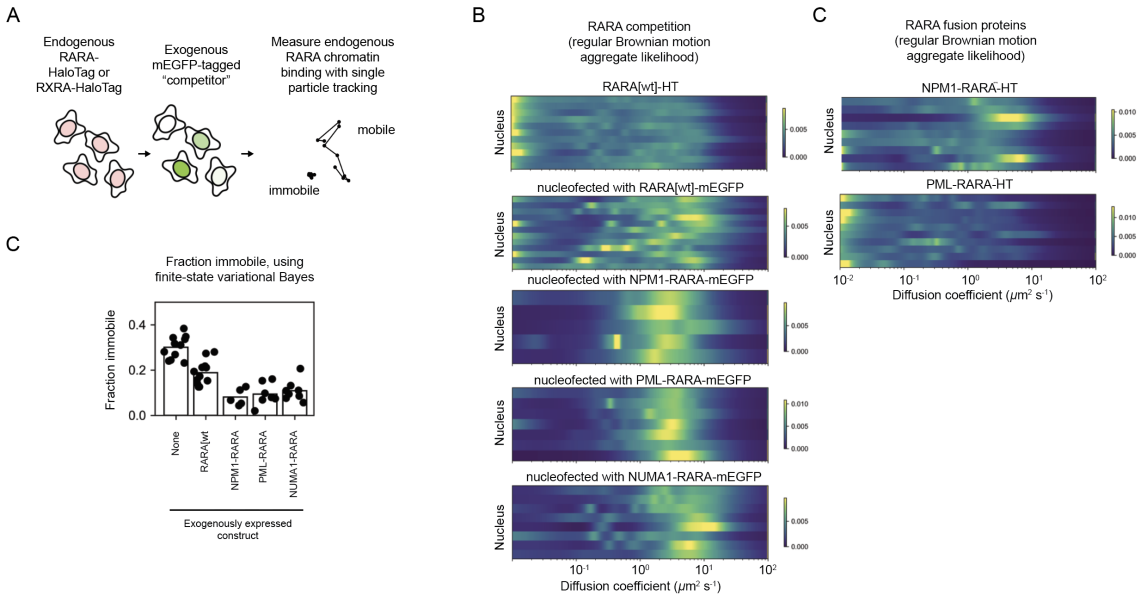

Figure 6.5: Live cell "competition" experiments with single particle tracking. (A) Schematic of the competition experiment. Endogenously tagged cell lines are transfected with GFP-bearing transgenes, and the resulting mixture is analyzed with spaSPT analysis modalities. (B) Aggregate regular Brownian motion likelihoods (localization error 30 nm) for endogenously expressed RARA-HT in the presence of overexpressed transgenes. The subplots on the right are transiently expressed PML-RARA-HaloTag and NPM1-RARA-HaloTag. (C) Aggregate regulate Brownian motion likelihoods for HaloTagged RARA fusion proteins in U2OS cells. Fusion protein were expressed by Lonza Amaxa II nucleofection. (D) Quantification of the fraction of immobile RARa-HT particles using a finite-state variational Bayes algorithm.

DNA binding (C88). In contrast, deletion of the unstructured NTD had no effect on the immobile fraction. Since RARA requires the LBD to form a heterodimer with RXR competent to bind DNA, these observations support the hypothesis that the origin of the immobile state in these experiments is DNA-bound RARA-HaloTag.

Interestingly, the aggregate likelihood method also revealed that the distribution of diffusion coefficients for either RARA[ΔLBD] or RARA[ΔDBD] was tighter than for the RARA[ΔNTD] or RARA[ΔCTD] constructs, with a near absence of slowly diffusing particles in the range <5.0 μm^2 s^{-1}. This suggests that slowly-diffusing states may correspond to the heterodimeric state, although further experiments are required to determine this. We remark that deletion of the unstructured CTD seemed to markedly reduce the immobile fraction while retaining slow-diffusing states. Since deletion of the unstructured CTD is expected to place HaloTag in close proximity to the LBD, further experiments analgous to Fig. 6.2B are required to determine that this construct is still function for activation of target genes.

Deletion of the DBD or LBD also increased fluorescence recovery after photobleaching (Fig. 6.4B). Combined with the spaSPT results, it is likely that the faster recovery is due to the loss of the immobile fraction along with the mobile state becoming associated with generally higher diffusion coefficients. Since most of the recovery occurs within 10 seconds for wildtype RARA-HaloTag, DNA binding times by RARA are expected to be faster than 10 seconds. We observed no effect of agonist treatment on recovery, indicating that exchange of corepressors for coactivators does not modify the diffusive behavior of RARA sufficiently to be observed in a FRAP experiment. Since changes in mobility have been observed by FCS [101], this may reflect the low sensitivity of the FRAP technique.

6.2.3 Assessing competition between RARA and exogenously expressed competitors

To determine how the diffusive behavior of RARA-HaloTag is affected by the presence of RARA fusion proteins derived from APL, we expressed three mEGFP-tagged fusion proteins in our RARA-HaloTag cell lines and assayed the endogenous RARA-HaloTag by spaSPT (Fig. 6.5A). The fusion proteins selected include the PML-RARA, most common fusion protein in APL, as well as the rarer variants NPM1-RARA and NUMA1-RARA. Expression of these fusion proteins impacted RARA-HaloTag dynamics as strongly as deletion of the DNA-binding domain itself (Fig. 6.5B). All three fusion proteins had a similar effect. In contrast, overexpression of wildtype RARA-mEGFP had a milder effect, resulting in a slight shift toward faster diffusion coefficients along with a reduction in the immobile fraction.

Curious about the diffusive behavior of the RARA fusion proteins themselves, we expressed HaloTagged NPM1-RARA and PML-RARA fusion proteins in wildtype U2OS nuclei (Fig. 6.5C). Interestingly, both fusion proteins exhibited distinct dynamics - PML-RARA is strongly biased toward slower diffusion coefficients and has a substantial immobile fraction, while NPM1-RARA has a much small immobile fraction that is only present in a subset of the nuclei assayed. Strangely, the profile for NPM1-RARA-HaloTag is unlike that of both RARA-HaloTag and endogenously tagged NPM1-HaloTag (Fig. 5.15).

The effect of the RARA fusion proteins on wildtype RARA was consistent with a competition mechanism. To determine if this was also the case for the effects of RXRA, we assayed the effect of RARA fusion protein expression on RXRA binding dynamics with FRAP (Fig. 6.6). While overexpression of wildtype RARA modestly reduced recovery, expression of the RARA fusion proteins had a much more pronounced effect, generally slowing RXRA recover (Fig. 6.6B). Moreover, the

ordinarily uniform distribution of nuclear RXRA-SNAPf is disrupted in these the presence of the fusion proteins. RXRA colocalizes with the small, speckle-like bodies exhibited by the PML-RARA-mEGFP transgene (Fig. 6.6C). Recovery in these speckles was markedly slower than in the nucleoplasm, and this effects also held in the case of NPM1-RARA for FRAP experiments performed in the nucleolus (Fig. 6.6B).

Because the RARA fusion proteins have apparently opposite effects on RARA-HaloTag and RXRA-SNAPf, these spaSPT experiments support a competition mechanism. However, experiments with endogenously tagged RXRA must be performed for corroboration.

6.2.4 Autoregulation of RARA expression levels

A puzzling aspect of these results was that the exogenously expressed RARA-mEGFP acted as an extremely weak competitor compared to the RARA fusion proteins (Fig. 6.5B, C). On one hand, this could be due to an increased affinity between RXRA and the RARA fusion proteins relative to the native RXRA-RARA interaction. On the other, it could arise from changes in the expression levels of these proteins. In the course of the spaSPT experiments, we had noticeably fewer detections for RARA-HT in the presence of exogenously expressed RARA-mEGFP, suggesting changing in the expression level of RARA-HT may be at least partly responsible for this effect.

To investigate this effect in more detail, we assayed the change in the expression level of endogenous RARA-HaloTag in response to exogenous expression of RARA- or RXRA-mEGFP (Fig. 6.7). RARA-HaloTag ordinarily exhibits extremely similar expression levels between cells (Fig. 6.2C, 6.2D), but in the presence of exogenous RARA-mEGFP the expression heterogeneity was increased, with a notable negative covariance between the levels of endogenous and exogenous RARA (Fig. 6.7B, C). Interestingly, this effect became stronger under atRA treatment, under which the entire pool of RARA contracts due to atRA-triggered degradation [102].

Expression of exogenous RXRA-mEGFP appeared to have the opposite effect (Fig. 6.7D), in both the presence and absence of atRA. The opposite effects of RARA and RXRA suggest the presence of a mode of heterodimerization-dependent autoregulation.

Analysis of the spatial distribution of exogenously expressed RARA-mEGFP showed an expression level-dependent localization (Fig. 6.7E). At low expression levels, RARA-mEGFP is predominately nuclear, while the distributions becomes less bi-

ased between the nucleus and cytoplasm at higher expression levels.

Known modes of autoregulation in the *RAR* genes fall into two categories (Fig. 6.7F). The three *RAR* genes each have two promoters. The second promoter often contains a RARE and is activated in response to retinoic acid [103]. The second mode of autoregulation is the agonist-induced degradation of RAR/RXR heterodimers [102]. Neither of these mechanisms can account for the expression level effects observed here, which occur in the absence of retinoic acid.

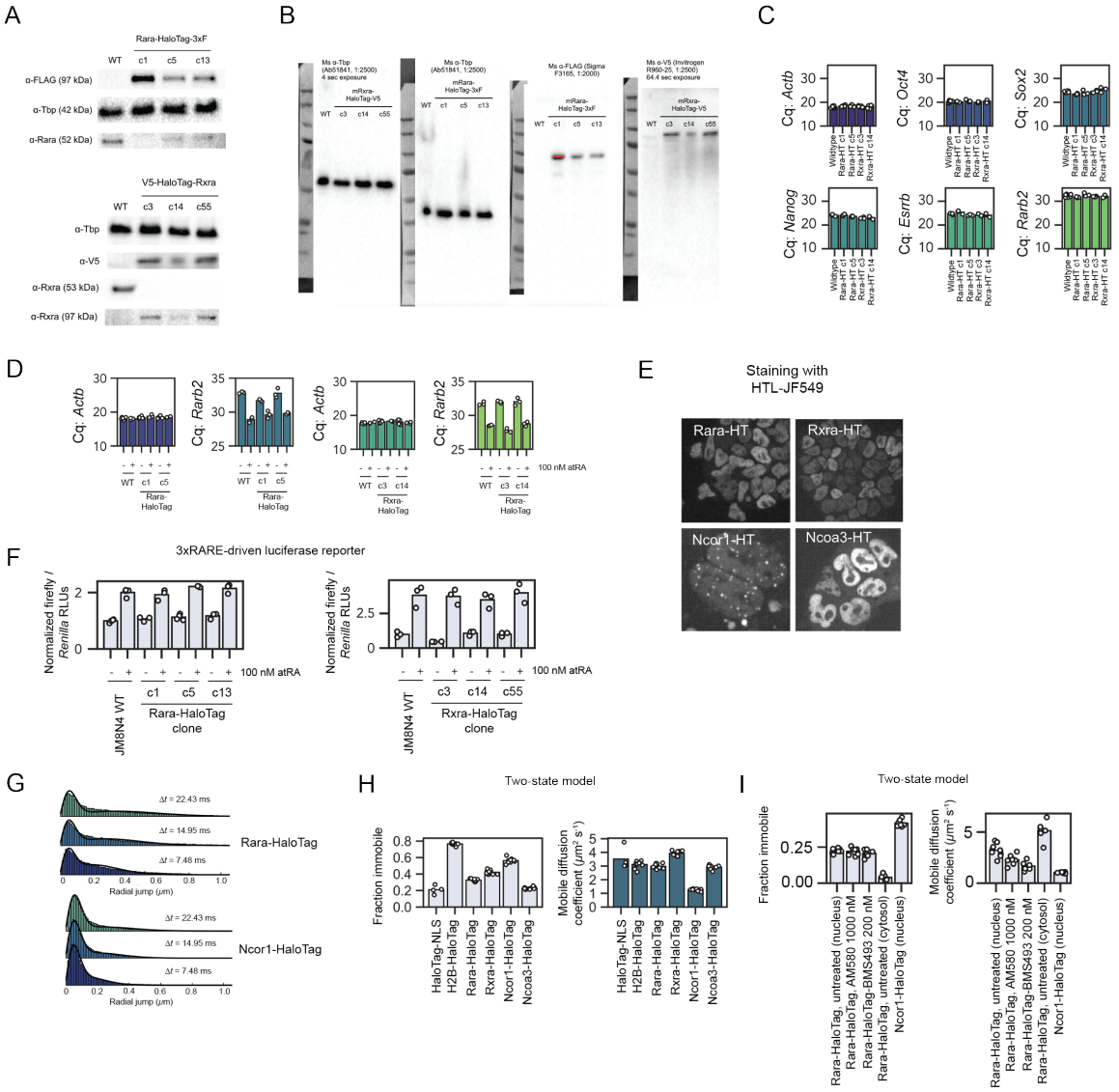

Figure 6.8: Endogenous tagging of RARA and coregulators in JM8N4 mouse embryonic stem cells. (A) Anti-FLAG and anti-V5 Western blots verifying the endogenous tag for Rara-HaloTag-3xFLAG and V5-HaloTag-Rxra in JM8N4 cells. Rara-HaloTag-3xFLAG clones c5 and c13 are heterozygotes. (B) Raw Western blots corresponding to (A). (C) RT-qPCR to assess expression of pluripotency-related genes in endogenously tagged Rara-HaloTag and HaloTag-Rxra cell lines. In the absence of RA or another agonist, *Rarb2* is not expressed. (D) RT-qPCR to assess response of *Rarb2* transcript to all-trans retinoic acid (atRA) in tagged cell lines. (E) Spinning disk confocal microscopy of tagged cell lines. (F) Luciferase assays confirming response of endogenously tagged cell lines to atRA. (G) spaSPT jump length distributions for tagged cell lines using the photoactivatable PA-JFX549 dye. (H) Quantification of diffusion modes for endogenously tagged Rara, Rxra, Ncor1, and Ncoa3 using two-state SpotOn [60]. (I) Same as (G), but in the presence of agonists/inverse agonists of retinoic acid receptor. Trajectories were spatially segregated into nuclear and cytoplasmic components and analyzed separately.

6.2.5 Mobility of Rara, Rxra, and coregulators in mouse embryonic stem cell nuclei

All-trans retinoic acid has a pronounced effect on mouse embryonic stem cells, inducing differentiation [104]. To understand how the mobility of Rara is affected by this process, we produced JM8N4 mouse embryonic stem cell lines with HaloTagged Rara, Rxra, Ncor1, and Ncoa3 (Fig. 6.8). Rara- and Rxra-tagged cell lines expressed pluripotency-related transcripts at levels indistinguishable from the parent population (Fig. 6.8C) and upregulated the *Rarb2* transcript in response to all-trans retinoic acid (Fig. 6.8D). Additionally, these cell lines activated expression of a RARE-driven luciferase reporter comparably to the parent cells (Fig. 6.8F). All four proteins localized to the nucleus, while subnuclear speckles apparent for Ncor1-HaloTag (Fig. 6.8E). Notably, the morphology of nuclei was aberrant for Ncoa3-HaloTag cells, and these cells generally have a reduced growth rate as well, probably indicating that the tagged Ncoa3 has reduced or no function.

To assay the dynamics of these proteins, we used a multi-state regular Brownian motion model with a jump histogram-based inference method (Fig. 6.8G). In all cases, the diffusion coefficient for one of the states converged to a value below 0.01 $\mu m^2\ s^{-1}$, which are effectively immobile molecules for the purpose of the spaSPT assay. Interestingly, while both Rara-HaloTag and Rxra-HaloTag exhibited nearly identical mobile diffusion coefficients, Ncor1-HaloTag was substantially slower (Fig. 6.8H) and also had a larger bound fraction.

Upon treatment with Am580, a strong synthetic retinoid analog, or BMS493, an inverse agonist that promotes association with corepressors [105], we observed no change in the immobile fraction of Rara-HaloTag. In contrast, treatment with both Am580 and BMS493 decreased the mobile diffusion coefficient of Rara-HaloTag. While agonists have been observed to slow the diffusion of RARA before [101], an effect attributed to the association of RARA with large coactivator complexes, the fact that BMS493 has a similar effect indicates that not all Rara-HaloTag molecules are in complex with corepressors in the absence of agonist as implied by Fig. 6.1A.

Revisiting these experiments with the techniques developed elsewhere in this thesis will prove useful for comparison with the U2OS RARA-HT experiments.

6.3 Discussion

The complexity of the T2NR regulatory network has made this system challenging to dissect. In vitro studies have identified a regulatory logic underlying binding site selection and heterodimerization among T2NRs that is not always replicated in

cells [106]. For this reason, measurements of T2NR dynamics in cells are valuable complements to our existing body of in vitro knowledge about these proteins.

An unexpected aspect of our results is the large heterogeneity in the observed modes of diffusion for RARA (Fig. 6.3). RARA appears to occupy diffusive states ranging from 0.1 to 10.0 $\mu m^2\ s^{-1}$. Mutation of the DNA-binding domain not only abolishes the immobile fraction, but also upshifts the mobile diffusion coefficients. This suggests that the slow modes of diffusion are dependent on DNA binding events, perhaps too fast to be observed with spaSPT. Indeed, diffusion of a particle in equilibrium between two diffusive states with sufficiently fast rates of conversion manifests as a single state with diffusion coefficient intermediate between that of the two states [43].

While further spaSPT studies with endogenously tagged RXR need to be conducted, our existing results support a competition-for-RXR mechanism. Indeed, the presence of RARA fusion proteins strongly reduces chromatin binding by endogenous RARA. This effect is comparable in magnitude to deletion of the RARA DNA-binding domain itself (Fig. 6.5). At the same time, the immobile fraction of RXR is increased (Fig. 6.6), and RXR colocalizes with the PML-RARA fusion protein. Together, these results support a mechanism wherein the RARA fusion proteins sequester endogenous RXR in long-lived immobile states, rendering endogenous RARA monomeric and unable to bind chromatin.

Wildtype RARA is a much weaker competitor than the RARA fusion proteins (Fig. 6.5C). We observed negative autoregulation of RARA expression levels on the timescale of a typical spaSPT experiment that may contribute to this effect (Fig. 6.7). In contrast to the known modes of *RARA* autoregulation, this effect was retinoic acid-independent, although it became more pronounced in the presence of atRA. Intriguingly, exogenously expressed RARA and RXR had opposite effects on the expression level of endogenous RARA. A possible explanation is that the monomeric RARA may be intrinsically less stable than the RARA/RXR heterodimer.

Finally, preliminary results in mouse embryonic stem cells provide insight into the relation of Rar/Rxr and coregulator dynamics. In the absence of agonist, T2NRs are believed to be associated with corepressor complexes NCoR and SMRT (Ncor1 and Ncor2, respectively), which dissociate upon agonist binding. We found that in the absence of agonist, Rar/Rxr and Ncor1 exhibit distinct modes of motion (Fig. 6.8). In particular, Ncor1 diffuses more slowly than Rar/Rxr. Biochemical studies have shown that Ncor1 exists in a large protein complex containing HDACs, which may account for this slow diffusion [107]. Agonist or inverse agonist treatment both reduced the mobility of Rar/Rxr, suggesting that in untreated mESCs, a substantial fraction free Rar/Rxr monomers and dimers are not associated with

either corepressor or activator. These results are consistent with an RXR FCS study in which agonist was observed to shift Rxr to a slower-moving state [108]. An interesting subject for future work is to identify whether the genomic binding profile of Rar and Rxr is modified by inverse agonist treatment in mESCs.

6.4 Materials and methods

Tissue culture. Human U2OS cells (female, 15 year old, osteosarcoma) were cultured under 5% CO_2 at 37 degrees C in DMEM containing 4.5 g/L glucose supplemented with 10% fetal bovine serum and 10 U/mL penicillin-streptomycin. Cells were subpassaged at a ratio of 1:6 every 3-4 days. The stable cell line expressing H2B-HaloTag-SNAPf was described previously [23] [24]. Expression of HaloTag and HaloTag-NLS were induced by nucleofection of PiggyBac vectors containing the proteins under EF1a promoters.

JM8N4 mouse embryonic stem cells [109] (RRID CVCL-J962; obtained from the KOMP Repository at UC Davis) were cultured on plates pre-coated with autoclaved 0.1% gelatin solution in feeder-free conditions. Culture medium was knockout DMEM supplemented with 15% fetal bovine serum and LIF: 500 mL knockout DMEM (ThermoFisher, Waltham, MA, 10829018), 6 mL MEM non-essential amino acids (ThermoFisher 11140050), 6 mL GlutaMax (ThermoFisher 35050061), 5 mL penicillin-streptomycin (ThermoFisher 15140122), 4.6 μL β-mercaptoethanol (Sigma-Aldrich M3148), 90 mL fetal bovine serum (HyClone, Logan, UT, FBS SH30910.03 lot AXJ47554) and in-house purified LIF. Medium on mES cells were changed daily and cells were subpassaged every 2 days.

For spaSPT experiments, cells were grown on 25 mm circular No. 1.5H coverglasses (Marienfeld, Germany, High-Precision 0117650) that had been sonicated in ethanol for 10 min, plasma-cleaned, then stored in isopropanol until use. U2OS cells were grown directly on the coverglasses in the regular culture medium. mES cells were grown on coverglasses that had been coated with Corning Matrigel matrix (Corning 354277; ThermoFisher 08-774-552) according to the manufacturer's instructions. For both U2OS and mES cells, the medium was changed immediately before imaging (after dye labeling) into phenol red-free mediujm to reduce background, while all other components of the medium remained unchanged.

Nucleofection. Because lipofection-based transfection methods often produce substantial background labeling in experiments with fluorescent dyes, for all imaging experiments involving exogenous expression we used the Lonza Amaxa II Nucleofector System with Cell Line Nucleofector Kit V reagent (Lonza VCA-1003). Briefly, U2OS cells were grown in 10 cm plates (ThermoFisher) for two days prior to

nucleofection, trypsinized, spun down at 1200 rpm for 5 min, combined with vector and Kit V reagent according to manufacturer's instructions, and nucleofected with program X-001 on an Lonza Amaxa II Nucleofector. After nucleofection, cells were immediately resuspended in regular culture medium at 37˚ C and plated.

CRISPR/Cas9-mediated gene editing. Endogenous tagging of RARA in U2OS cells and Rara, Rxra, Ncor1, and Ncoa3 in mES cells were performed with a protocol roughly following [23] with some modifications. This protocl relies on FACS sorting for cells that have been correctly modified to express HaloTag fused to the target protein. Briefly, for U2OS cells, we nucleofected cells with plasmid expressing 3xFLAG-SV40NLS-pSpCas9 from a CBh promoter [111], mVenus from a PGK promoter, and guide RNA from a U6 promoter, along with a second plasmid encoding the homology repair donor. The homology repair donor was built in a pUC57 backbone modified to contain HaloTag (and/or 3xFLAG/V5, as relevant) with $\sim$500 base pairs of homologous genomic sequence on either side. Synonymous mutations were introduced at the cut site to prevent repeated targeting by Cas9. Three distinct guide RNAs were used for each target, which were nucleofected into separate populations of cells to be pooled for subsequent analysis. 24 hours after the initial nucleofection, we screened for mVenus-expressing cells using FACS and pooled these mVenus-positive cells in 10 cm plates. 5 days after plating, we labeled cells with HTL-TMR (Promega G8251) and screened for TMR-positive, mVenus-negative cells. Cells were diluted to single clones and plated in 96-well plates for a 2-3 week outgrowth step, during which the medium was replaced every 3 days. The 96-well plates were then screened for wells containing single colonies of U2OS cells, which were split by manual passage into two replicate wells in separate 96-well plates. One of these replicates was used to subpassage, while the other was used to harvest genomic DNA for PCR and sequencing-based screening for the correct homology repair product. In PCR screens, we used three primer sets: (A) primers external to HaloTag, expected to amplify both the wildtype allele and the edited allele, (B) a primer internal to HaloTag and another external to it on the 5' side, expected to amplify only the edited allele, and (C) a primer internal to HaloTag and another external to it on the 3' side, expected to amplify only the edited allele. PCR products were gel-purified (Qiagen 28704) and sequenced; only clones with the target sequence were kept for continued screening. Finally, we selected a subset of these clones to confirm the expression of the HaloTagged allele by Western blot.

For mES cells, we used the same general strategy. Lipofectamine 3000 was used to transfect Cas9 plasmid [111] and homology repair donor into a wildtype JM8N4 population. One day later, cells were sorted for mVenus, replated, and at the next passage were labeled with TMR-HTL, and sorted for TMR-positive, mVenus-negative cells. These cells were plated at low density on gelatin-coated 15 cm

plates. Clones were then picked and expanded for screening as described for U2OS cells.

Western blots. Antibodies were as follows. The ratio indicate the dilution factors used for Western blot. human TBP, Abcam Ab51841, 1:2500 (mouse); FLAG, Sigma-Aldrich F3165, 1:2000 (mouse); V5, Invitrogen R960-25, 1:2500 (mouse); mRxra, Abcam 125001, 1:500 (mouse); mRara, Abcam Ab41934, 1:400 (mouse); anti-Mouse HRP, Invitrogen 31430, 1:5000 (goat).

For Western blots, cells were collected by scraping from plates in ice-cold PBS, then pelleted. Cell pellets were resuspended in lysis buffer (0.15 M NaCl, 1% NP-40, 50 mM Tris-HCl (pH 8.0), and a cocktail of protease inhibitors (Sigma-Aldrich 11697498001)), agitated for 30 min at 4˚ C, then centrifuged for 20 min at 12000 rpm, 4˚ C. The supernatant was then mixed with 2x Laemmli (to final 1x), boiled for 5 min, then run on 12.5% SDS-PAGE. After transfer to nitrocellulose, the membrane was blocked with 10% condensed milk in TBST (500 mM NaCl, 10 mM Tris-HCl (pH 7.4), 0.1% Tween-20) for one hour at room temperature. Antibodies were suspended in 5% condensed milk in TBST at the dilutions indicated above and incubated, rocking at 4˚ C overnight. After hybridization, the membrane was washed three times for 10 min with TBST at room temperature, hybridized with an anti-mouse HRP secondary antibody in 5% condensed milk in TBST for 60 min at room temperature, washed three more times with TBST for 10 min, then visualized with Western Lightning Plus-ECL reagent (PerkinElmer NEL103001) according to manufacturer instructions and imaged on a Bio-Rad ChemiDoc imaging system. Different exposure times were used for each target.

Luciferase assays. All luciferase assays used pGL3-RARE-luciferase, a reporter containing firefly luciferase driven by an SV40 promoter with three retinoic acid response elements (RAREs). pGL3-RARE-luciferase was a gift from T. Michael Underhill (Addgene plasmid 13458 ; http://n2t.net/addgene:13458 ; RRID:Addgene_13458) [110]. Luciferase assays were performed on cells cultivated in 6-well plates; cells were transfected with 100 ng pGL3-RARE-luciferase and 10 ng pRL Renilla (Promega E2261) using Mirus TransIT-2020 Transfection Reagent (Mirus MIR 5404) for U2OS cells or Lipofectamine 3000 (ThermoFisher L3000015) for mES cells. Transfection was performed one day before assaying luciferase expression with the Dual-Luciferase Reporter Assay System (Promega E1910) according to manufacturer's instructions. Readout was performed on a GloMax luminometer (Promega).

RT-qPCR. Total RNA was purified from cell pellets with RNeas Plus Mini kit (Qiagen) and quantified by UV absorption (Nanodrop). We used 1 μg of total RNA for reverse transcription with iScript reverse transcription supermix (Bio-Rad 1708840).

qPCR was performed with SYBR Select Master Mix for CFX (Applied Biosystems, ThermoFisher) on a Bio-Rad CFX Real-Time PCR system.

Cell labeling. For spaSPT and FRAP experiments, cells were labeled with one of two methods, depending on the type of dye. For non-photoactivatable fluorescence dyes including TMR-HTL (tetramethylrhodamine-HaloTag ligand; Promega G8251) and SNAPtag-JF646 (a generous gift from Luke Lavis; [14]), we stained cells with 100 nM dye in regular culture medium for 10-20 min, then performed three 10 min incubations in dye-free culture medium separated by PBS washes. All PBS and culture medium was kept at 37˚ C during washes.

For experiments with photoactivatable dyes, which have lower cell permeability and slower wash in/wash out kinetics, we labeled cells with 100 nM dye in regular culture medium for 30 min, followed by four 30 min incubations in dye-free culture medium at 37˚ C. Between each incubation, we washed twice with PBS at 37˚ C. After the final incubation, cells were changed into phenol red-free medium for imaging.

spaSPT. spaSPT experiments were performed with a custom-bulit Nikon TI microscope equipped with a 100X/NA 1.49 oil-immersion TIRF objective (Nikon apochromat CFI Apo TIRF 100X Oil), an EMCCD camera (Andor iXon Ultra 897), a perfect focus system to account for axial drift, an incubation chamber maintaining a humidified 37˚ C atmosphere with 5% CO_2, and a laser launch with 405 nm (140 mW, OBIS, Coherent), 488 nm, 561 nm, and 633 nm (all 1 W, Genesis Coherent) laser lines. Laser intensities were controlled by an acousto-optic Tunable Filter (AA Opto-Electronic, AOTFnC-VIS-TN) and triggered with the camera TTL exposure output signal. Lasers were directed to the microscope by an optical fiber, reflected using a multi-band dichroic (405 nm/488 nm/561 nm/633 nm quadband, Semrock) and focused in the back focal plane of the objective. The angle of incident laser was adjusted for highly inclined laminated optical sheet (HiLo) conditions [13]. Emission light was filtered using single band-pass filters (Semrock 593/40 nm for PAJFX549 and Semrock 676/37 nm for PAJF646). Hardware was controlled with the Nikon NIS-Elements software.

For stroboscopic illumination, the excitation laser (561 nm or 633 nm) was pulsed for 1-2 ms (most commonly 1.5 ms) at maximum (1 W) power at the beginning of the frame interval, while the photoactivation laser (405 nm) was pulsed during the $\sim$447 μs camera transition time, so that the background contribution from the photoactivation laser is not integrated. For all spaSPT, we used an EMCCD vertical shift speed of 0.9μs and conversion gain setting 2. On our setup, the pixel size after magnification is 160 nm and the photon-to-grayscale gain is 109. 15000-30000 frames with this sequence were collected per nucleus, during which

the 405 nm intensity was manually tuned to maintain low density of fluorescent particles per frame.

Localization and tracking. To produce trajectories from raw spaSPT movies, we used a custom tracking tool publicly available on GitHub (quot). All localization and tracking was performed with the following settings:

- *Detection*: generalized log likelihood ratio test with a 2D Gaussian kernel of fixed radius 190 nm (detection method `llr` with `k = 1.2`, a 15 pixel window size (`w = 15`), and a log ratio threshold of 16.0 (`t = 16.0`).
- *Localization*: Gauss-Newton estimation of a 2D integrated Gaussian point spread function model (localization methood `ls_int_gaussian`) with fixed radius 190 nm, window size 9 pixels, maximum 20 iterations per PSF, with a damping term of 0.3 for parameter updates.
- *Tracking*: Method `conservative` with a 1.2 μm search radius.

After localization and tracking, all trajectories in the first 1000 frames of each movie were discarded. Localization density tends to be high in these frames, so they can contribute tracking errors that compromise accuracy. The mean localization density for most movies in the remaining set of frames is less than one emitter per frame.

For experiments involving HaloTag or HaloTag-NLS, which move quickly, we used a broader search radius at 2.5 μm. All other settings were kept the same.

FRAP. For fluorescence recovery after photobleaching experiments, we used an inverted Zeiss LSM 710 AxioObserver confocal microscope equipped with a motorized stage, an incubation chamber maintaining 37˚ C and 5% CO_2, a heated stage, an X-Cite 120 illumination source and a 561 nm laser line. All FRAP experiments used the TMR 561 nm laser line and a 40X Plan NeoFluar NA1.3 oil-immersion objective at a 100 nm pixel size. 20 frames were acquired before bleaching a circular spot with the 561 nm microscope. In all cases, the spot was chosen with radius 13 pixels and the pixel dwell time of the laser was slowed to its minimum setting to maximize bleaching. In these experiments, the bleaching time is 1-2 seconds, meaning that dynamics below this timescale are not accessible.

For FRAP analysis, we used a custom Python pipeline to integrate the intensity of the bleach spot at each frame while accounting for cell drift. Recovery was quantified by normalizing the integrated intensity of the spot to its average intensity in the 20 pre-bleach frames. We normalized for photobleaching by dividing the

recovery curve by the normalized sum nuclear intensity at each frame.

Spinning disk confocal imaging. Experiments using spinning disk confocal imaging were performed at the UC Berkeley High-Throughput Screening Facility on a Perkin Elmer Opera Phenix equipped with a controller for 37˚ C and 5% CO_2, using a built-in 40X water immersion objective.

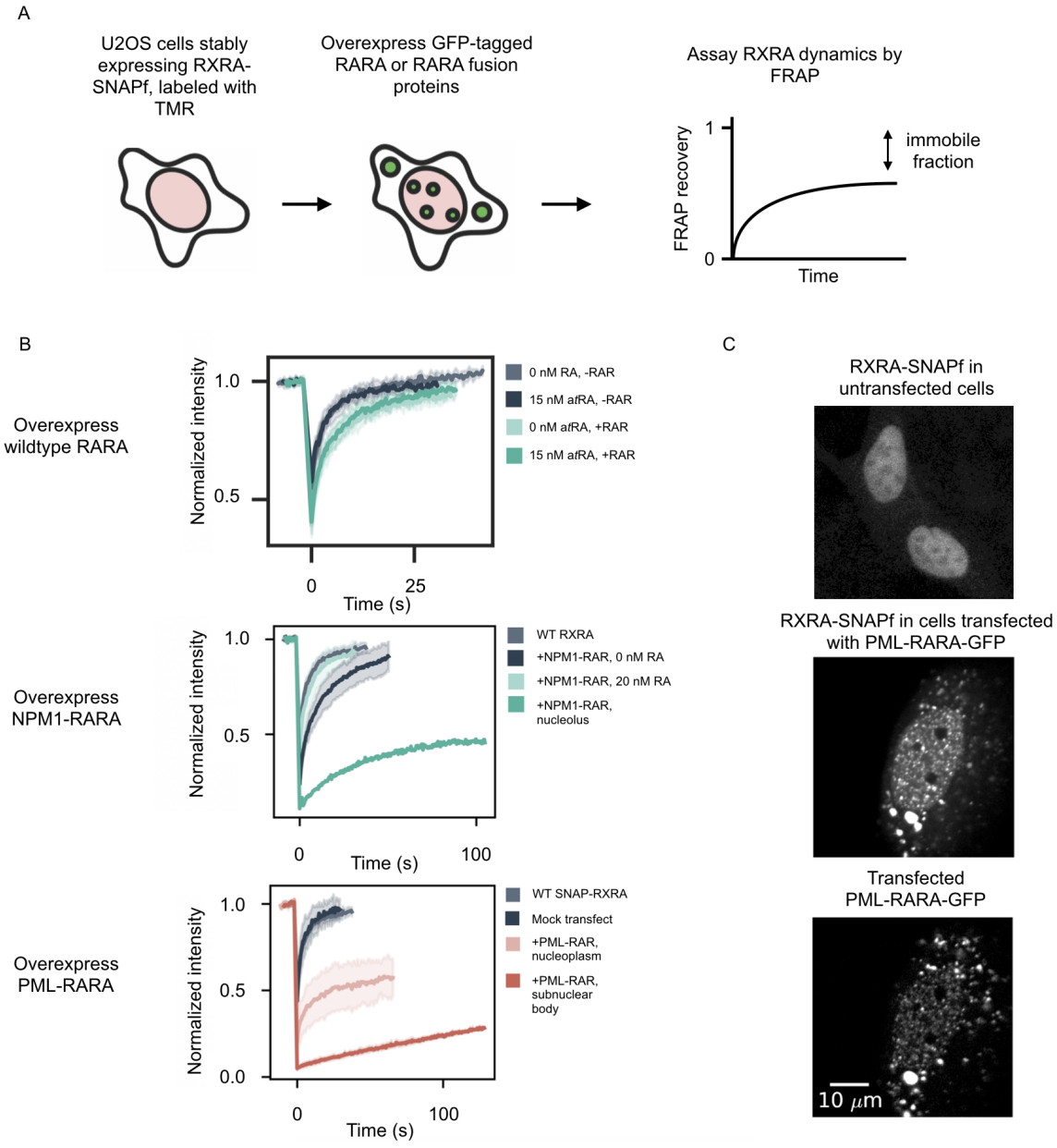

Figure 6.6: Influence of exogenously expressed RARA and RARA fusion proteins on RXRA binding dynamics. (A) Schematic of the assay. Cells bearing a stably expressed RXRA-SNAPf are transfected with mEGFP-tagged RARA or RARA fusion proteins. Cells with GFP expression are then selected for FRAP experiments. (B) FRAP results. In all cases, the recovery curves represent FRAP experiments for the same stably expressed RXRA-SNAPf U2OS cell line. atRA is all-trans retinoic acid. In the case of the lower subplot, "subnuclear body" refers to the small speckle-like bodies produced by expression of PML-RARA (see (C)). (C) Confocal microscopy images. The upper two subplots are stably expressed RXRA-SNAPf stained with JF646 (Methods) and the lower subplot is an mEGFP PML-RARA transgene.

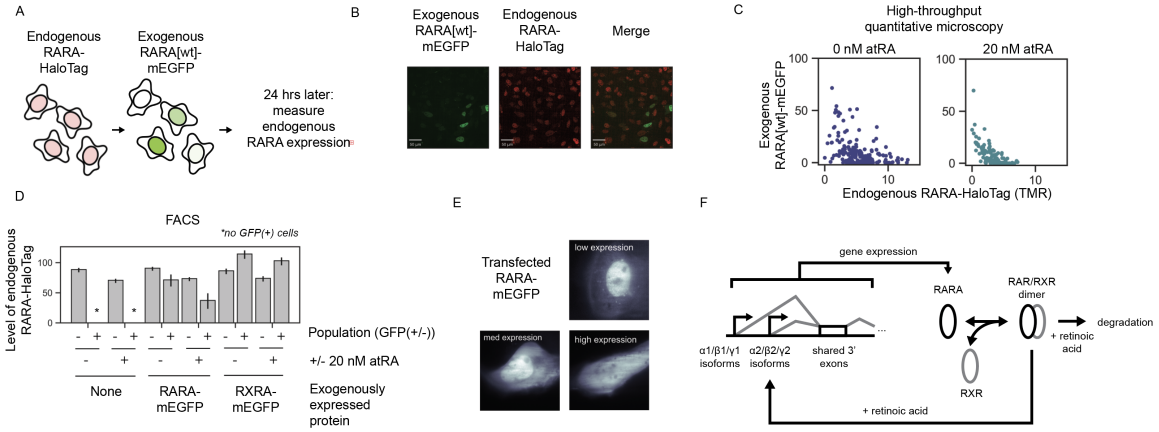

Figure 6.7: Influence of exogenously expressed RARA on endogenously RARA expression levels. (A) Schematic of the experiment. Cells bearing endogenously tagged RARA-HaloTag-3xFLAG are labeled with TMR, then are made to express exogenous mEGFP-tagged RARA by nucleofection. 24 hours later, expression levels are assayed. (B) Example spinning disk microscopy image of exogenous and endogenous RARA in a population of U2OS cells. (C) Quantification of the results in (B) using a high-througphut spinning disk microscope. The levels of exogenous and endogenous RARA on the axes are the integrated intensities across each nucleus. (D) Quantification of the same experiment using FACS. Cells were analyzed in two populations - mEGFP fluorescent cells (which contain the transgene) and mEGFP(-) cells. Fluorescence was quantified relative to a wildtype U2OS control labeled with TMR. (E) Sample widefield fluorescence microscopy images of RARA-mEGFP transgene at different expression levels. (F) Cartoon depicting the known RAR autoregulatory logic. Note that the only the α2, β2, and γ2 promoters are responsive to retinoic acid activation.

Appendix A

Appendix: Spot detection with generalized log likelihood ratio tests and related variance-normalized detection algorithms.

Spot detection is often the first step in the treatment of raw data generated in spaSPT. Here, we outline a method for detection with an arbitrary "spot model" - taking the place of a PSF in traditional fixed-cell PALM/STORM - based on a general log likelihood ratio test (GLRT). The development that follows is strongly inspired by Stephen Kay's book [81] as well as the GLRT described in Sergé and coworkers' tracking algorithm [112].

From an algorithmic perspective, many detection methods operate along the same general lines as the GLRT method: an image is convolved with one or more kernels, which are combined in some way to yield a modified image that is subsequently thresholded to identify spots. A critical factor in these algorithms is the final threshold that defines the detection criterion. Classic spot detection methods such as difference-of-Gaussian (DoG) or Laplacian-of-Gaussian (LoG) filtering, require tuning this threshold to the particular intensity and camera noise characteristics of each dataset.

The advantage of the GLRT is that it is *invariant* with respect to the intensity of the original image. As a result, a user can apply a threshold on data collected in different experiments, or even on different cameras, and expect the same general behavior of the algorithm.

As it turns out, the GLRT can be phrased in a particularly convenient algorithmic form in terms of image convolutions that allows us to transfer this intensity-

invariant property to a much broader class of image detection methods, including classic DoG and LoG filters.

A.1 Generalized likelihood ratio tests for spot detection in 2D images

Consider the problem of spot detection in a noisy 2D image. Say we start with the following:

1. We have an image A with shape $M \times N$ pixels, so that A_{ij} is the observed intensity of the i^{th}, j^{th} pixel (with $i \in \{1, 2, ..., M\}$, $j \in \{1, 2, ..., N\}$).
2. We have a spot model S with dimensions $m \times n$, so that S_{ij} is the model intensity for the i^{th}, j^{th} pixel (with $i \in \{1, 2, ..., m\}$, $j \in \{1, 2, ..., n\}$).
3. The spot model is normalized so that $\sum_i \sum_j S_{ij} = 1$.

A common choice for S is a 2D Gaussian. Then we have

$$S_{ij} \propto \exp\left(-\frac{(i - \frac{m+1}{2})^2 + (j - \frac{n+1}{2})^2}{2r_0^2}\right)$$

Normalization of this density on $i \in \{1, 2, ..., m\}$, $j \in \{1, 2, ..., n\}$ yields the spot model. (Because we are working with discrete pixels - this is *not* equivalent to the usual 2D Gaussian normalization factor $(2\pi r_0^2)^{-1}$.)

Here, r_0 defines the width of the Gaussian kernel. We assume that r_0, and whatever other parameters that define S for other models, are constant.

Divide the image A into subregions $A^*_{y_c,x_c}$ that are each centered on a pixel (y_c, x_c), so that $A^*_{i,j} = A_{y_c - \frac{m+1}{2} + i, x_c - \frac{n+1}{2} + j}$. (When m and n are not odd, the division in the indices can be assumed to give the division floor, so that $\frac{8+1}{2} = 4$.)

Then for every choice of (y_c, x_c), consider two hypotheses, H_0 and H_1:

H_0. The neighborhood A^* contains only normally distributed noise, so that

$$A^*_{ij} = b_0 + N_{ij}$$
$$N_{ij} \sim \mathcal{N}\left(0, \sigma_0^2\right)$$

where b_0 is a background term.

H_1. There is a spot centered at (y_c, x_c), so that

$$A^*_{ij} = IS_{ij} + b_1 + N_{ij}$$
$$N_{ij} \sim \mathcal{N}\left(0, \sigma_1^2\right)$$

where I is the spot intensity, b_1 is a background term, and N_{ij} is normally distributed noise with variance σ_1^2.

The approach for the GLRT is first to identify the maximum likelihood of each hypothesis given the observed A^*. This requires implicitly finding the maximum likelihood parameters for each hypothesis, which we'll denote as $\hat{m}_0$, $\hat{\sigma}_0^2$ (for hypothesis H_0) and $\hat{m}_1$, $\hat{\sigma}_1^2$, $\hat{I}$ (for hypothesis H_1). Then the log likelihood ratio of the two hypothesis is compared to a given threshold T. If the log ratio exceeds T, then we call a spot centered on (y_c, x_c). Otherwise, the neighborhood A^* is not considered for subsequent analysis.

Maximum likelihood parameters for hypothesis H_0

First consider hypothesis H_0, which models the observed A^* data as normally distributed noise on top of some background term. Then the likelihood of parameters m_0, σ_0^2 given the observed spot profile A^* is

$$\begin{aligned} \mathcal{L}\left[m_0, \sigma_0^2 \mid A^*\right] &= \prod_{i,j} \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left(-\frac{(A^*_{ij} - b_0)^2}{2\sigma_0^2}\right) \\ \log \mathcal{L}\left[m_0, \sigma_0^2 \mid A^*\right] &= -\frac{nm}{2} \log\left(2\pi\sigma_0^2\right) - \frac{1}{2\sigma_0^2} \sum_{ij} \left(A^*_{ij} - b_0\right)^2 \end{aligned} \tag{A.1}$$

We seek the maximum likelihood estimate $\hat{b}_0$. Differentiating A.1 with respect to b_0 and setting this equal to zero,

$$\left.\frac{\partial \log \mathcal{L}\left[b_0, \sigma_0^2 \mid A^*\right]}{\partial b_0}\right|_{\hat{b}_0} = \frac{1}{\sigma_0^2} \sum_{ij} \left(A^*_{ij} - \hat{b}_0\right) = 0$$
$$\hat{b}_0 = \frac{1}{nm} \sum_{ij} A^*_{ij}$$

which is just the sample mean of A^*. Solving similarly for $\hat{\sigma}_1^2$,

$$\begin{aligned}\left.\frac{\partial \log \mathcal{L}\left[\hat{b}_0, \sigma_0^2\right]}{\partial \sigma_0^2}\right|_{\hat{\sigma}_0^2} &= -\frac{nm}{2\hat{\sigma}_0^2} + \frac{1}{\hat{\sigma}_0^4}\sum_{ij}\left(A_{ij}^* - \hat{b}_0\right)^2 \\ &= 0 \\ \implies \hat{\sigma}_0^2 &= \frac{1}{nm}\sum_{ij}\left(A_{ij}^* - \hat{b}_0\right)^2\end{aligned}$$

which is the sample variance of A^*. Now that we have maximum likelihood estimates for b_0 and σ_0^2, define the log likelihood of hypothesis H_0 as the log likelihood evaluated at these estimates:

$$\log \mathcal{L}\left[H_0 \mid A^*\right] = \log \mathcal{L}\left[\hat{b}_0, \hat{\sigma}_0^2 \mid A^*\right]$$

Plugging this back into the log likelihood A.1, we obtain

$$\log \mathcal{L}\left[H_0 \mid A^*\right] = -\frac{nm}{2}\left(1 + \log\left(2\pi\hat{\sigma}_0^2\right)\right) \tag{A.2}$$

Maximum likelihood parameters for hypothesis H_1

The maximum likelihood parameters for hypothesis H_1 are $\hat{m}_1$, $\hat{\sigma}_1^2$, and $\hat{I}$. We can obtain these using similar methods to above, albeit with added complexity due to the spot model. Our log likelihood function is now

$$\log \mathcal{L}\left[b_1, \sigma_1^2, I \mid A^*\right] = -\frac{nm}{2}\log\left(2\pi\sigma_1^2\right) - \frac{1}{2\sigma_1^2}\sum_{ij}\left(A_{ij}^* - b_1 - IS_{ij}\right)^2 \tag{A.3}$$

First, differentiate the log likelihood with respect to σ_1^2 and evaluate at the ML estimate $\hat{\sigma}_1^2$:

$$\begin{aligned}\left.\frac{\partial \log \mathcal{L}\left[b_1, \sigma_1^2, I \mid A^*\right]}{\partial \sigma_1^2}\right|_{\hat{\sigma}_1^2} &= -\frac{nm}{2\hat{\sigma}_1^2} + \frac{1}{2\hat{\sigma}_1^4}\sum_{ij}\left(A_{ij}^* - b_1 - IS_{ij}\right)^2 \\ &= 0 \\ \hat{\sigma}_1^2 &= \frac{1}{nm}\sum_{ij}\left(A_{ij}^* - b_1 - IS_{ij}\right)^2\end{aligned} \tag{A.4}$$

Substituting this back into the log likelihood A.3, we obtain

$$\log \mathcal{L}\left[b_1, \hat{\sigma}_1^2, I \mid A^*\right] = -\frac{nm}{2}\left(1 + \log\left(2\pi\hat{\sigma}_1^2\right)\right) \tag{A.5}$$

Now seek $\hat{b}_1$ by differentiating A.5:

$$\left.\frac{\partial \log \mathcal{L}\left[b_1, \sigma_1^2, I \mid A^*\right]}{\partial b_1}\right|_{\hat{b}_1} = -\frac{nm}{2\hat{\sigma}_1^2}\left.\frac{\partial \hat{\sigma}_1^2}{\partial b_1}\right|_{\hat{b}_1} = 0$$
$$\implies \hat{b}_1 = \frac{1}{nm}\sum_{ij}\left(A_{ij}^* - IS_{ij}\right)$$

Plugging this back into A.4, we have

$$\begin{aligned}
\hat{\sigma}_1^2 &= \frac{1}{nm}\sum_{ij}\left(A_{ij}^* - \frac{1}{nm}\sum_{i'j'}\left(A_{i'j'}^* - IS_{i'j'}\right) - IS_{ij}\right)^2 \\
&= \frac{1}{nm}\sum_{ij}\left(\left(A_{ij}^* - \langle A^*\rangle\right) - I\left(S_{ij} - \langle S\rangle\right)\right)^2 \\
&= \frac{1}{nm}\sum_{ij}\left(\tilde{A}_{ij}^* - I\tilde{S}_{ij}\right)^2
\end{aligned}$$

where in the last line we have defined $\tilde{A}_{ij}^* = A_{ij}^* - \langle A^*\rangle$ and $\tilde{S}_{ij} = S_{ij} - \langle S\rangle$, the mean-subtracted image and model intensities. Notice that when $I = 0$, we recover the maximum likelihood estimator for $\tilde{\sigma}_1^2$ under H_0, which is just the sample variance.

Now, the only variable that remains is $\hat{I}$. We seek $\hat{I}$ such that

$$\begin{aligned}
\left.\frac{\partial \log \mathcal{L}\left[\hat{b}_1, \hat{\sigma}_1^2, \hat{I}\right]}{\partial I}\right|_{\hat{I}} &= -\frac{nm}{2\hat{\sigma}_1^2}\left.\frac{\partial \hat{\sigma}_1^2}{\partial I}\right|_{\hat{I}} = 0 \\
&= -\frac{2}{nm}\sum_{ij}\left(\tilde{A}_{ij}^* - \hat{I}\tilde{S}_{ij}\right)\tilde{S}_{ij} \\
&= \frac{2}{nm}\left(-\sum_{ij}\tilde{A}_{ij}^*\tilde{S}_{ij} + \hat{I}\sum_{ij}\tilde{S}_{ij}^2\right) \\
\hat{I} &= \frac{\sum_{ij}\tilde{A}_{ij}^*\tilde{S}_{ij}}{\sum_{ij}\tilde{S}_{ij}^2} = \frac{1}{\xi}\sum_{ij}\tilde{A}_{ij}^*\tilde{S}_{ij}
\end{aligned}$$

where in the last equation we have let $\xi = \sum_{ij}\tilde{S}_{ij}^2$, a constant for each choice of model S. Now the maximum likelihood estimate for the variance under H_1 be-

comes

$$\begin{aligned}
\hat{\sigma}_1^2 &= \frac{1}{nm}\sum_{ij}\left(\tilde{A}_{ij}^* - \hat{I}\tilde{S}_{ij}\right)^2 \\
&= \frac{1}{nm}\sum_{ij}(\tilde{A}_{ij}^*)^2 - \frac{2\hat{I}}{nm}\sum_{ij}\tilde{A}_{ij}^*\tilde{S}_{ij} + \frac{\hat{I}^2}{nm}\sum_{ij}\tilde{S}_{ij}^2 \\
&= \frac{1}{nm}\sum_{ij}(\tilde{A}_{ij}^*)^2 - \frac{2\hat{I}^2\xi}{nm} + \frac{\hat{I}^2}{nm}\sum_{ij}\tilde{S}_{ij}^2 \\
&= \frac{1}{nm}\sum_{ij}(\tilde{A}_{ij}^*)^2 - \frac{\xi}{nm}\hat{I}^2 \\
&= \hat{\sigma}_0^2 - \frac{\xi}{nm}\hat{I}^2
\end{aligned}$$

Log likelihood ratio of hypotheses H_0 and H_1

With $\hat{\sigma}_0^2$ and $\hat{\sigma}_1^2$ in hand, we can evaluate the log likelihood ratio of the two hypotheses. Suppose, as before, that we have a subregion A^* of an image A centered at the point (y_c, x_c). Then, using A.2 and A.5, the log likelihood of H_1 relative to H_0 reduces to a comparison of the variances:

$$\begin{aligned}
\log\mathcal{L}\left[y_c, x_c\right] &= \log\mathcal{L}\left[H_1 \mid A^*\right] - \log\mathcal{L}\left[H_0 \mid A^*\right] \\
&= \left(-\frac{nm}{2}\left(1 + \log\left(2\pi\hat{\sigma}_1^2\right)\right)\right) - \left(-\frac{nm}{2}\left(1 + \log\left(2\pi\hat{\sigma}_0^2\right)\right)\right) \\
&= -\frac{nm}{2}\log\left(\frac{\hat{\sigma}_1^2}{\hat{\sigma}_0^2}\right) \\
&= -\frac{nm}{2}\log\left(\frac{\hat{\sigma}_0^2 - \frac{\xi}{nm}\hat{I}^2}{\hat{\sigma}_0^2}\right) \\
&= -\frac{nm}{2}\log\left(1 - \frac{\xi\hat{I}^2}{nm\hat{\sigma}_0^2}\right) \\
&= -\frac{nm}{2}\log\left(1 - \frac{\left(\sum_{ij}\tilde{A}_{ij}^*\tilde{S}_{ij}\right)^2}{nm\xi\hat{\sigma}_0^2}\right)
\end{aligned}$$

Recognizing that $\sum_{ij} \tilde{A}^*_{ij}\tilde{S}_{ij} = \sum_{ij} A^*_{ij}\tilde{S}_{ij}$ and that

$$\begin{aligned}\hat{\sigma}_0^2 &= \frac{1}{nm}\sum_{ij}\left(A^*_{ij} - \langle A^*\rangle\right)^2 \\ &= \frac{1}{nm}\left(\sum_{ij}(A^*_{ij})^2 - 2\langle A^*\rangle\sum_{ij}A^*_{ij} + nm\langle A^*\rangle^2\right) \\ &= \frac{1}{nm}\left(\sum_{ij}(A^*_{ij})^2 - \frac{1}{nm}\left(\sum_{ij}A^*_{ij}\right)^2\right),\end{aligned}$$

where $\tilde{U}_{ij} = \frac{1}{nm}$ for all pixels (i, j).

we can rewrite this log likelihood ratio as

$$\log \mathcal{L}\left[y_c, x_c\right] = -\frac{nm}{2}\log\left[1 - \frac{\left(\sum_{ij}A^*_{ij}\tilde{S}_{ij}\right)^2}{\xi\left(\sum_{ij}(A^*_{ij})^2 - \frac{1}{nm}\left(\sum_{ij}A^*_{ij}\right)^2\right)}\right] \tag{A.6}$$

This is now solely in terms of the observed subimage A^* and the model S. We call a spot centered at (y_c, x_c) when

$$\log \mathcal{L}\left[A^*\right] \geq T$$

for some log likelihood threshold T.

A.2 Algorithms to compute the GLRT on sequences of images

Equation A.6 can be put into a computationally tractable form using image convolutions. Here, we'll use $A * K$ to denote the convolution of the image A with a kernel K, so that if K has dimensions $n \times m$,

$$(A * K)_{ij} = \sum_{i'=1}^{n}\sum_{j'=1}^{m} S_{i'j'}A^*_{i+\frac{m+1}{2}-i', j+\frac{n+1}{2}-j'}$$

Then we note that, if U is a uniform kernel of shape $m \times n$ where $U_{ij} = 1/nm$ for all i, j,

$$\sum_{ij} (A^*_{ij})^2 = nm \left(A^2 * U\right)_{ij}$$

$$\left(\sum_{ij} A^*_{ij}\right)^2 = nm \left(A * U\right)^2_{ij}$$

where $A^2 = A \circ A$ is the Hadamard product of A with itself.

Then we can rewrite the GLRT as

$$\log \mathcal{L}\left[y_c, x_c\right] = -\frac{nm}{2} \log \left[1 - \frac{\left(A * \tilde{S}\right)^2_{y_c, x_c}}{\xi \left(nm(A^2 * U)_{y_c, x_c} - (A * U)^2_{y_c, x_c}\right)}\right] \tag{A.7}$$

Again, our criterion for detection is that this log ratio exceed some threshold T. Forming this inequality and rearranging, we can eliminate the logarithm to yield an equivalent criterion for detection:

$$\frac{\left(A * \tilde{S}\right)^2_{ij}}{nm \left(A^2 * U\right)_{ij} - \left(A * U\right)^2_{ij}} \geq \xi \left(1 - e^{-2T/nm}\right) \tag{A.8}$$

Now the right-hand side is a constant for a particular choice of S and T.

Because the left-hand side of equation A.8 can be computed for every point (y_c, x_c) in the original image in a small number of vectorizable operations (convolution, multiplication, subtraction, and division), it lends itself readily to a fast detection algorithm (Algorithm A.1).

The numerator $(A * \tilde{S})^2$ will be equal and positive for spots of both positive and negative curvature. Since we generally do not want spots with positive curvature (which are "holes" in the image), it is useful to filter on the curvature by setting the ratio in A.8 to zero for all pixels (i, j) such that $(A * \tilde{S})_{ij} < 0$.

Finally, it is useful to add a morphological closing step with a circular structuring element to the binary image after thresholding for spots. This makes it less likely that a single spot generates multiple detections.

Algorithm A.1: Generalized log likelihood ratio test for a sequence of images

Parameters:

- An image stack $(A)_{t\in\{1,2,...\}}$ with image dimensions $N \times M$
- A spot model S with dimensions $m \times n$
- A log likelihood ratio for detection T

Precompute:

- $\overline{S}$, the Fourier transform of the mean-subtracted kernel $S - \langle S \rangle$ padded with zeroes to $N \times M$
- $\overline{U}$, the Fourier transform of a uniform kernel padded with zeroes to $N \times M$
- The normalization term $\xi = \sum_{ij} \left(S_{ij} - \langle S \rangle\right)^2$
- The adjusted detection threshold $T^* = \xi \left(1 - e^{-2T/mn}\right)$

Algorithm: For each image frame $t = 1, 2, ...$:

1. Calculate the Fourier transforms $\overline{A} = \mathcal{F}\left[A_t\right]$ and $\overline{A^2} = \mathcal{F}\left[A_t^2\right]$.
2. Generate the convolution $I = \mathcal{F}^{-1}\left[\overline{A} \circ \overline{S}\right]$.
3. Generate the convolution $B = nm\mathcal{F}^{-1}\left[\overline{A^2} \circ \overline{U}\right]$.
4. Generate the convolution $C = \mathcal{F}^{-1}\left[\overline{A} \circ \overline{U}\right]$.
5. Generate the binary image $L = [(I \circ I)/(B - C) \geq T^*]$, where the division and subtraction operations are taken to apply elementwise.
6. Set $L_{ij} =$ false for all i, j such that $I_{ij} < 0$.
7. Call spots in pixels (i, j) such that L_{ij} is true.

A.3 Other variance-normalized spot detection algorithms

Examining the left-hand side of equation A.8, one can see that it operates as a kind of signal-to-noise metric:

$$\frac{\left(A * \tilde{S}\right)^2}{nm\left(A^2 * U\right) - \left(A * U\right)^2} \sim \xi^2 \frac{(\text{signal})^2}{\text{local variance}} \tag{A.9}$$

The factor ξ^2 appears since the maximum likelihood estimator for the intensity of a spot centered at (y_c, x_c) is $\hat{I}_{y_c,x_c} = \frac{1}{\xi}\left(A * \tilde{S}\right)_{y_c,x_c}$.

As discussed previously, the tremendous utility of the GLRT comes from the invariance of the intensity threshold T with respect to the absolute intensity of the images on which it is run. In other words, one threshold can be used for images with very different intensities. Equation A.9 provides an intuitive reason for this property. We can see that the GLRT operates by:

1. for every pixel, evaluate the maximum likelihood estimate for the intensity of a spot centered on that pixel
2. for every pixel, evaluate the local variance in a box surrounding that pixel
3. compare the ratio of (1) and (2)

Spot detection methods like DoG or LoG, which work by convolving the image with a mean-zero kernel, correspond to the "signal" part of the ratio in A.9. (These convolutions differ from the maximum likelihood estimator for the spot intensity, $\hat{I}$, by a factor of ξ.) Simply by substituting either a DoG or LoG kernel for S, we can obtain viable GLRTs. The resulting detection methods work similarly to naive DoG or LoG, but inherit the intensity invariance characteristic of the GLRT. This can make them much more reliable for high-throughput image processing, a setting in which the user cannot manually examine all images to check that the detection method is both sensitive and accurate.

There are also numerous spot detection methods that do not operate by simple convolution of the image with a kernel. These include, for example, methods based on the local Hessian determinant. While such methods do not transfer in a straightforward manner to the GLRT framework, many still share the general principle of thresholding a function defined on the original image. Departing from the GLRT, we can use A.9 as inspiration for a very simple "variance-normalization" method.

Suppose, again, that A is a $M \times N$ image. We'll assume that we have a spot detection method that returns I, an $M \times N$ array such that the value of I_{ij} is related to the likelihood that there is a spot centered on (i, j) in the original image.

Choose some odd integer $\alpha > 1$. Define $U_\square$ as a "hollow" square kernel of shape $m \times n$ (where m and n are both odd) such that

$$(U_\square)_{ij} = \begin{cases} 0 & \text{if} \quad \left|i - \frac{m+1}{2}\right| \leq \frac{\alpha-1}{2} \text{ and} \\ & \quad \left|j - \frac{n+1}{2}\right| \leq \frac{\alpha-1}{2}, \\ 1 & \text{otherwise} \end{cases}$$

Then we propose the following normalization:

$$\bar{I} = \frac{I^2}{mn\,(I^2 * U_\square) - (I * U_\square)^2} \tag{A.10}$$

$\bar{I}$ is now in terms of the ratio of the local SNR of the original spot detection method. The method penalizes spots that have magnitude similar to the magnitude of local noise. As a result, the method safeguards against spurious detection of spots in noisy neighborhoods.

The normalization A.10 is not perfect in that we're missing the factor ξ, which accounts for the variance contributed by the choice of spot model. Nevertheless, A.10 operates fairly well in practice for detection methods such as the Hessian determinant (see main text).

Algorithm A.2: Simple variance-normalized spot detectors

Parameters:

- an image A of shape $M \times N$
- a spot detection algorithm, represented by the operator $D[\cdot]$
- the variance window size n, an odd integer
- the spot width α, an odd integer

Precompute:

- $\overline{U}$, the Fourier transform of the hollow kernel padded with zeroes to $N \times M$

Algorithm: For each image frame $t = 1, 2, ...$,

1. Run the normal spot detection method $I = D[A]$.
2. Compute the convolution $A = \mathcal{F}^{-1}\left[\mathcal{F}[I]\,\overline{U}\right]$
3. Compute the convolution $B = \mathcal{F}^{-1}\left[\mathcal{F}\left[I^2\right]\overline{U}\right]$
4. Calculate the normalized image $\bar{I} = I^2/(n^2A - B^2)$
5. Apply a spot detection threshold T

Appendix B

Appendix: Gaussian processes

Both regular Brownian motion (RBM) and fractional Brownian motion (FBM) are instances of Gaussian processes and inherit the powerful inferential techniques associated with these processes. Here we briefly review Gaussian processes, focusing on the cases of RBM and FBM. We also describe the "modified diffusion coefficient" $\overline{D}$, which is useful in practical situations involving FBM.

B.1 Definition

A Gaussian process X_t for $t \in \mathbb{R}$, $t \geq 0$ is a stochastic process such that, for any finite collection of indices t_1, t_2, ..., t_n, the vector $\mathbf{X} = (X_{t_1}, X_{t_2}, ..., X_{t_n})$ has a multivariate normal PDF.

Because the multivariate normal density is completely described by its second-order statistics, the specification of a mean function $\mu(t)$ and a covariance function $\text{Cov}(t, s)$ is sufficient to completely define a Gaussian process.

Imagine that we evaluate the mean and covariance functions on a specific set of indices $t_1, ..., t_n$, corresponding to the vector $\mathbf{X} = (X_{t_1}, ..., X_{t_n})$. For convenience we'll usually denote this mean as $\mu_{\mathbf{X}}$ and the covariance matrix as $\Sigma_{\mathbf{X}}$ so that

$$\mu_{\mathbf{X}} = \begin{bmatrix} \mathbb{E}[X_{t_1}] \\ \mathbb{E}[X_{t_2}] \\ ... \\ \mathbb{E}[X_{t_n}] \end{bmatrix} = \begin{bmatrix} \mu(t_1) \\ \mu(t_2) \\ ... \\ \mu(t_n) \end{bmatrix}$$

$$\Sigma_{\mathbf{X}} = \begin{bmatrix} \text{Cov}(t_1, t_1) & ... & \text{Cov}(t_1, t_n) \\ ... & ... & ... \\ \text{Cov}(t_n, t_1) & ... & \text{Cov}(t_n, t_n) \end{bmatrix}$$

When considering the process at two possibly disjoint sets of indices - for instance, $\mathbf{X}_1 = (X_{t_1}, ..., X_{t_n})$ and $\mathbf{X}_2 = (X_{t_{n+1}}, ..., X_{t_m})$ - we'll extend the covariance function

to vectorial arguments such that

$$\text{Cov}\left(\mathbf{X}_1, \mathbf{X}_2\right) = \Sigma_{12} = \begin{bmatrix} \text{Cov}(t_1, t_{n+1}) & \ldots & \text{Cov}(t_1, t_m) \\ \ldots & \ldots & \ldots \\ \text{Cov}(t_n, t_{n+1}) & \ldots & \text{Cov}(t_n, t_m) \end{bmatrix}$$

Note that $\Sigma_{12} = \Sigma_{21}^T$.

B.2 Properties of Gaussian processes

Suppose that we have a discrete set of indices $t_1, ..., t_m$ that we've divided into two groups. The first group runs from t_1 to t_n and the second runs from t_{n+1} to t_m. Define $\mathbf{X}$ as the vector of a Gaussian process evaluated at the complete set of indices so that $\mathbf{X} = (X_{t_1}, ..., X_{t_m})$. Likewise, define $\mathbf{X}_1$ and $\mathbf{X}_2$ as the vector of the process evaluated at the first and second groups of indices respectively, so that $\mathbf{X}_1 = (X_{t_1}, ..., X_{t_n})$ and $\mathbf{X}_2 = (t_{n+1}, ..., t_m)$.

Then $\mathbf{X}$ has a multivariate normal density with the mean vector

$$\mu_{\mathbf{X}} = \begin{bmatrix} \mu_{\mathbf{X}_1} \\ \mu_{\mathbf{X}_2} \end{bmatrix}$$

and the block covariance matrix

$$\Sigma_{\mathbf{X}} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

The vector $\mathbf{X}$ inherits the properties associated with multivariate normal random vectors, the most useful of which we summarize below.

B.2.1 Independence

$\mathbf{X}_1$ and $\mathbf{X}_2$ are independent if and only if $\text{Cov}(\mathbf{X}_1, \mathbf{X}_2) = \mathbf{0}$.

B.2.2 Sum of two independent multivariate normal vectors

If $\mathbf{X}_1$ and $\mathbf{X}_2$ are independent and equal in cardinality, then their sum has the multivariate normal density

$$\mathbf{X}_1 + \mathbf{X}_2 \sim \mathcal{N}\left(\mu_{\mathbf{X}_1} + \mu_{\mathbf{X}_2}, \Sigma_{11} + \Sigma_{22}\right) \tag{B.1}$$

B.2.3 Marginal distributions

The marginal distributions of $\mathbf{X}_1$ or $\mathbf{X}_2$ are also multivariate normal random vectors:

$$\begin{aligned} \mathbf{X}_1 &\sim \mathcal{N}\left(\mu_{\mathbf{X}_1}, \Sigma_{11}\right) \\ \mathbf{X}_2 &\sim \mathcal{N}\left(\mu_{\mathbf{X}_2}, \Sigma_{22}\right) \end{aligned} \tag{B.2}$$

B.2.4 Conditional distributions

The conditional distribution of $\mathbf{X}_2$ given $\mathbf{X}_1$ is another multivariate normal random vector:

$$\mathbf{X}_2 \,\Big|\, \mathbf{X}_1 \sim \mathcal{N}\left(\mu_{\mathbf{X}_2} + \Sigma_{21}\Sigma_{11}^{-1}\left(\mathbf{X}_1 - \mu_{\mathbf{X}_1}\right),\ \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right) \tag{B.3}$$

Equation B.3 is the basis for most of the inference techniques associated with Gaussian processes in machine learning, since it enables the user to predict the value of the process at new indices given some previous observations.

B.2.5 Conditioning in the presence of measurement error

Suppose, as in B.3, that we have observed $\mathbf{X}_1 = (X_{t_1}, ..., X_{t_n})$ and we wish to predict the values of the process at a different set of indices, $\mathbf{X}_2 = \left(X_{t_{n+1}}, ..., X_{t_m}\right)$.

Equation B.3 assumes that we know the value of $\mathbf{X}_1$ *exactly*. More often, our measurement has some error associated with it. We'll imagine that this error is a Gaussian white noise process E_t, so that for $\mathbf{E} = (E_{t_1}, ..., E_{t_n})$ we have

$$\mathbf{E} \sim \mathcal{N}\left(0, \nu^2 I\right)$$

where I is the identity matrix. Using B.1, we then have

$$\mathbf{X}_1 + \mathbf{E} \sim \mathcal{N}\left(\mu_{\mathbf{X}_1}, \Sigma_{11} + \nu^2 I\right)$$

So we can simply apply B.3 while substituting $\Sigma_{11} + \nu^2 I$ for Σ_{11}.

B.3 Regular and fractional Brownian motion

Fractional Brownian motion can be defined as a Gaussian process with the mean function $\mu(t) = 0$ and the covariance function

$$\text{Cov}(t, s) = D\left(|t|^{2H} + |s|^{2H} - |t - s|^{2H}\right) \tag{B.4}$$

where $D \geq 0$ and $0 < H < 1$. We'll refer to D as the *diffusion coefficient* and H as the *Hurst parameter*. D governs the magnitude of the displacements of the motion, while H governs the memory effects. In particular, when $H < 1/2$, the process is subdiffusive and its increments (when measured at a discrete set of timepoints) are anticorrelated. Likewise, when $H > 1/2$, the process is superdiffusive and the increments are positively correlated. In the special case $H = 1/2$, the increments are uncorrelated and we have regular Brownian motion with the covariance

$$\text{Cov}(t, s) = 2D \min(t, s) \tag{B.5}$$

This leads to the familiar RBM mean squared displacement $\text{Var}(t) = 2Dt$.

Figure B.1 shows some sample FBM trajectories. The position of each process has been observed at ten discrete time points, and we demonstrate the use of conditioning property B.3 to obtain error bounds on the position at the rest of the time points. Note that the high amount of local noise in the subdiffusive FBM ($H = 0.3$) results in larger error bounds, even quite close to the observed points.

B.4 Modified diffusion coefficient $\overline{D}$

FBMs are useful because - in theory - they allow us to *separately* parametrize the magnitude of the spatial increments (via D) and the correlations between increments (via H). The magnitude of the spatial increments produced from FBM as defined in equation B.4, however, are highly dependent on both H and D.

It's easy to see this from the FBM mean squared displacement:

$$\text{MSD}_H(t) = \text{Cov}_H(t, t) = 2D\,|t|^{2H}$$

Suppose that $t = 0.005$ s (a typical spaSPT frame interval) and $D = 1.0\ \mu\text{m}^2\ \text{s}^{-2H}$. Then the root mean squared displacement of a process after one frame interval with $H = 0.3$ is 0.289 μm, while the same measurement for a process with $H = 0.5$ is 0.100 μm, and for $H = 0.7$ it becomes 0.035 μm.

This is nearly an order of magnitude variation in the mean jump length contributed by the Hurst parameter alone, which makes it difficult to experimentally compare diffusion coefficients when using FBM models. There is no general way to normalize the diffusion coefficient so that the Hurst parameter does not exert an effect on the spatial variance, since it goes contrary to the definition B.4. However, we can modify the diffusion coefficient to make comparisons easier at a particular timescale.

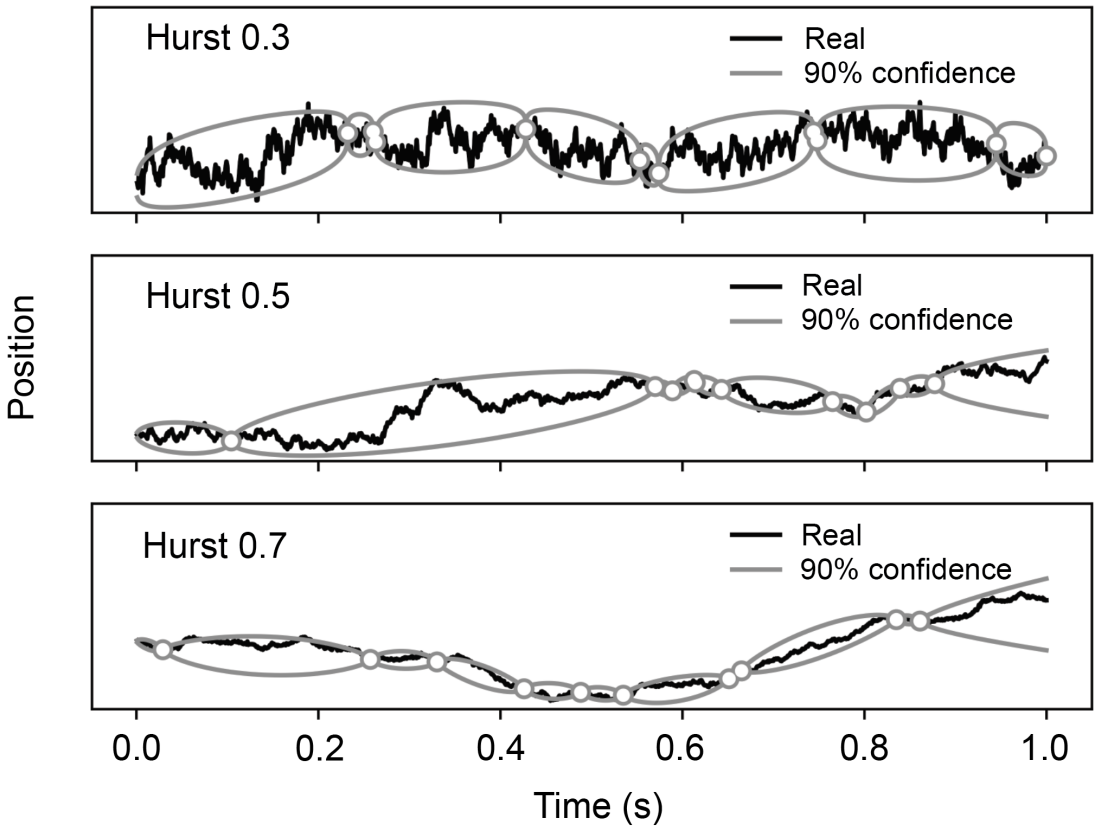

Figure B.1: Examples of fractional Brownian motion trajectories with different Hurst parameters. The upper process, with $H = 0.3$, corresponds to subdiffusion; the middle process, with $H = 0.5$, corresponds to regular Brownian motion; the lower process, with $H = 0.7$, corresponds to superdiffusion. Open circles indicate the points at which the process has been observed, and the gray lines indicate 90% confidence intervals obtained with the conditioning property B.3. These simulations make use of the modified diffusion coefficient $\overline{D}$.

In particular, let Δt be our frame interval, so that our observations are at times 0, Δt, $2\Delta t$, and so on. Then define the modified diffusion coefficient

$$\overline{D}_{\Delta t} = D\Delta t^{2H-1} \tag{B.6}$$

$\overline{D}$ has units of $\mu\text{m}^2\ \text{s}^{-1}$ and is defined so that $\text{MSD}_H(\Delta t) = 2\overline{D}\Delta t$, regardless of the value of H. This facilitates comparisons of the diffusion coefficient at the frame interval of interest. The two diffusion coefficients are easily converted using equation B.6. Figure B.2 demonstrates the effect of the modified diffusion coefficient on the MSDs for some FBMs. Notice that without the modification, the variance of FBMs with a low Hurst parameter undergoes an extremely large increase at early time values, making comparison between diffusion coefficients all but impossible.

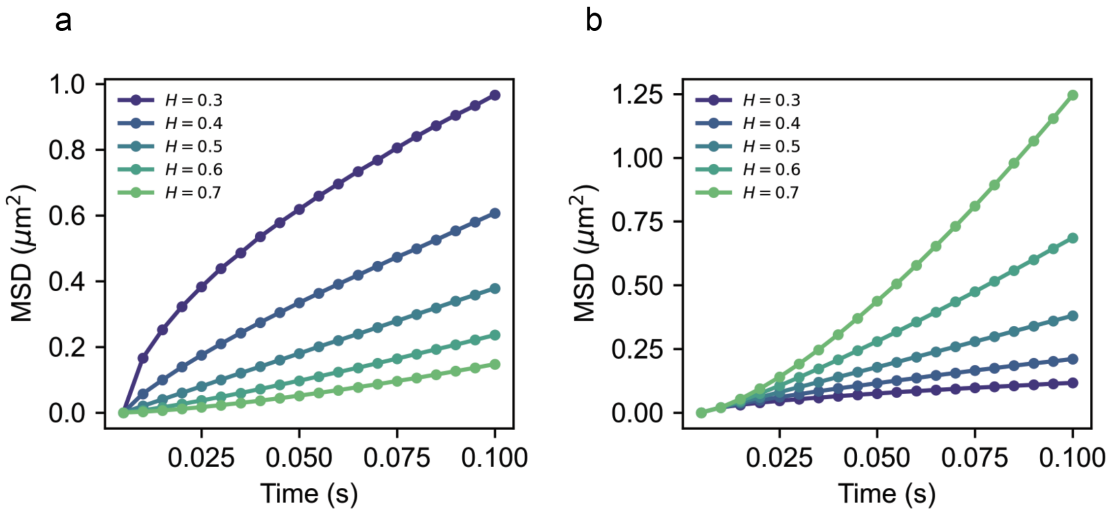

Figure B.2: Mean squared displacements of FBMs with various diffusion coefficients, using 5 ms frame intervals. Each line corresponds to a simulation with 100000 separate trajectories. (a) Using the unmodified diffusion coefficient $D = 2.0\ \mu\text{m}^2\ \text{s}^{-2H}$. (b) Using the modified diffusion coefficient $\overline{D} = 2.0\ \mu\text{m}^2\ \text{s}^{-1}$.

B.5 Alternative constructions of FBM

Here we have defined FBM by appealing to the covariance function of a zero-mean Gaussian process, but this is not the only way. The original constructions by Paul Levy [53] and Mandelbrot & Van Ness [51] followed a fundamentally different route. We feel this alternative route provides insight into the nature of FBM, and so we examine some of the analytical properties of this representation in this section. Our goal is not to present a mathematically rigorous view of FBMs and fractional calculus, but to introduce some informal relations that help to build intuition about the nature of FBM.

Suppose that N_t is a Gaussian white noise process so that $N_t \sim \mathcal{N}\left(0, \nu^2\right)$ and

$$\text{Cov}\left(N_t, N_s\right) = \begin{cases} \nu^2 & \text{if} \quad t = s \\ 0 & \text{otherwise} \end{cases}$$

As Kiyoshi Ito observed, the Wiener process B_t can be considered as the integral of this process:

$$B_t = \int_{-\infty}^{t} \mathrm{d}N_s$$

where the integral is defined in the sense of Ito [52]. Using the connection be-

tween the Fourier transform and integration, we have

$$\tilde{B}_k = -\frac{\tilde{N}_k}{ik}$$

where k is the frequency coordinate and $\tilde{B}_k$, $\tilde{N}_k$ are the Fourier transforms of B_t, N_t. The properties of Gaussian white noise require that $\tilde{N}_k$ is also a Gaussian white noise process - that is, it has equal intensity on all frequencies. Vitally, this means that the Wiener process can be understood as the sum of random frequencies (drawn from $\tilde{N}_k$) with amplitudes that are damped by $1/k$. (Equivalently, the power spectrum is damped by $1/k^2$.)

Since multiplication in Fourier space corresponds to convolution in real space, this implies the existence of a kernel $S(t)$ such that $\mathcal{F}[S] = -1/ik$ and so that B_t can be obtained by the convolution of N_t with $S(t)$.

A kernel satisfying these criteria is a modified signum function:

$$S(t) = \frac{1}{2}(\text{sgn}(t) + 1) = \begin{cases} 0 & \text{if} \quad t < 0 \\ 1 & \text{if} \quad t > 0 \\ \frac{1}{2} & \text{otherwise} \end{cases}$$

(The identity $\mathcal{F}[S] = -1/ik$ for $k \neq 0$ can be obtained by noting that $\mathrm{d}(\text{sgn})/\mathrm{d}t = 2\delta(t)$ where $\delta(t)$ is the delta function, then integrating by parts.)

This means that the Wiener process can be represented by the convolution

$$B_t = \int_{-\infty}^{\infty} S(t-s)\mathrm{d}N_s$$

One can verify that, substituting the definition of $S(t)$, we recover Ito's integral.

So far, this is just an overly complicated way to write an integral. The interesting part comes when we consider other ways to damp the white noise spectrum, which will produce types of random motion distinct from the Wiener process. Suppose that, rather than damping the frequencies by $(-ik)^{-1}$ as above, we instead damp them by $(-ik)^{\alpha-1}$, with some $\alpha \in \mathbb{R}$. Specifically, consider the transfer function

$$\tilde{S}^{(H)}(k) = \begin{cases} (-ik)^{H-\frac{3}{2}} & \text{if} \quad k \neq 0 \\ 0 & \text{if} \quad k = 0 \end{cases} \tag{B.7}$$

which corresponds to a real-space *Weyl kernel*

$$S^{(H)}(t) = \frac{\text{sgn}(t) + 1}{2|t|^{H-\frac{1}{2}}}$$

We've neglected some integration constants because they are inconsequential to our discussion. Notice that when $H = 1/2$, we recover the damping term $(-ik)^{-1}$, corresponding to the Wiener process. The Weyl kernel is the basis for the integral transform known as the *Weyl fractional integral*.

We are now in a position to appreciate the constructions by Levy and Mandelbrot/Van Ness. Inspired by a similar line of reasoning as the one that led to equation B.7, Paul Levy [53] considered the process defined by

$$X_t^{(H)} = \frac{1}{\Gamma(H+\frac{1}{2})} \int_0^t \frac{\mathrm{sgn}(t-s)\mathrm{d}N_s}{(t-s)^{H-\frac{1}{2}}} = \frac{1}{\Gamma(H+\frac{1}{2})} \int_0^t \frac{\mathrm{d}B_s}{(t-s)^{H-\frac{1}{2}}}$$

The main problem here is that Levy somewhat arbitrarily chose the lower limit of integration to be 0, which corresponds to a Riemann-Liouville fractional integral (rather than a Weyl integral) with base point 0. The gamma function emerges when considering the integration constants we have neglected.

Mandelbrot & Van Ness, who were the first to examine this process in detail, kept the spirit of Levy's construction intact while replacing the Riemann-Liouville integral with a Weyl fractional integral, which is very simply the convolution of a white noise process with the Weyl kernel:

$$X_t^{(H)} = \frac{1}{\Gamma(H+\frac{1}{2})} \left(\int_{-\infty}^0 \left((t-s)^{H-\frac{1}{2}} - (-s)^{H-\frac{1}{2}} \right) \mathrm{d}B_s + \int_0^t (t-s)^{H-\frac{1}{2}} \mathrm{d}B_s \right)$$

Our goal here is not to go into the intricacies of comparing the Riemann-Liouville and Weyl fractional integrals. In our opinion, both Levy's and Mandelbrot/Van Ness's constructions provide less intuition than the Weyl transfer function B.7 itself, which illustrates the following:

- When $H = 1/2$, Brownian motion is produced by damping the amplitudes of randomly chosen frequencies by k^{-1}.
- When $H < 1/2$, then we damp the amplitudes of randomly chosen frequencies by $k^{-\alpha}$ with $\alpha < 1$. As a result, the higher frequencies have a stronger influence - and the lower frequencies have a weaker influence - on the motion than for Brownian motion.
- When $H > 1/2$, then we damp the amplitudes of randomly chosen frequencies by $k^{-\alpha}$ with $\alpha > 1$. The lower frequencies have a stronger influence on the motion than for Brownian motion.

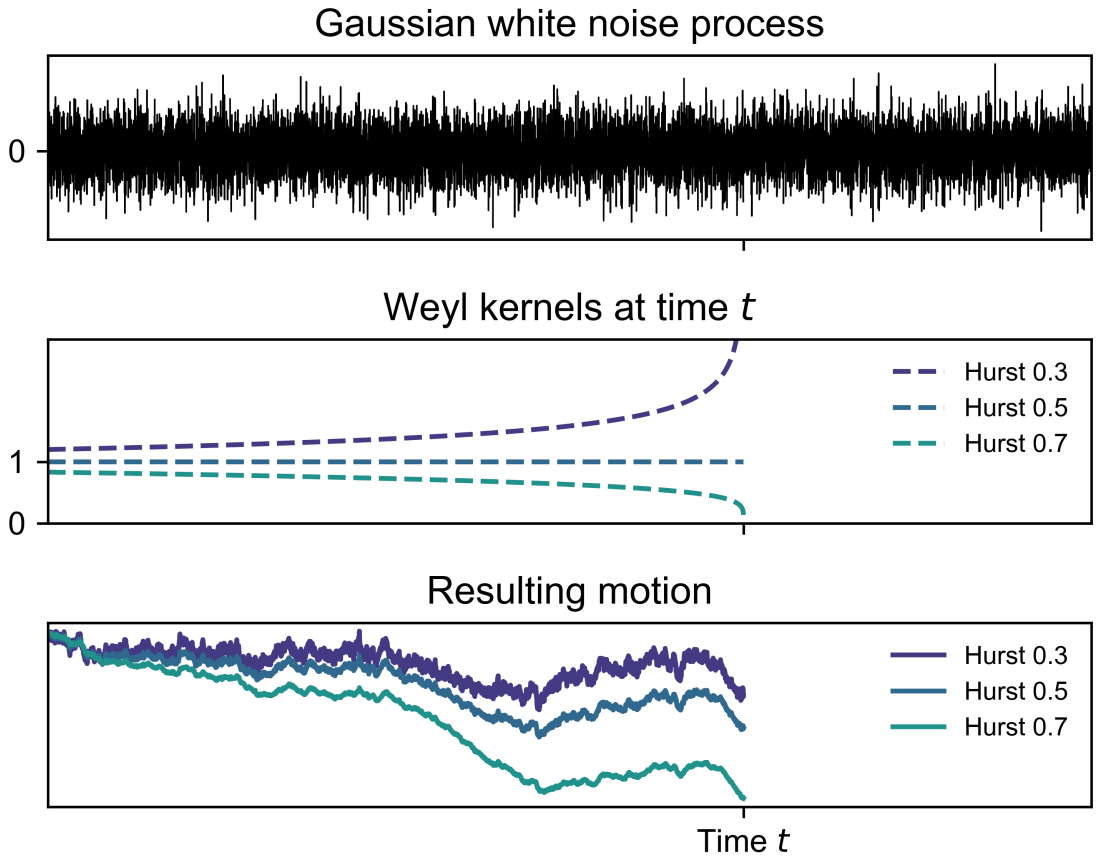

Figure B.3: Illustration of the construction of an FBM by convolution of white noise with the Weyl kernel.
The same white noise process (sampled at $6 \cdot 10^{-5}$ second timepoints) was convolved with Weyl kernels corresponding to subdiffusion ($H = 0.3$), regular Brownian diffusion ($H = 0.5$), and superdiffusion ($H = 0.7$). The second panel shows the Weyl kernels at a single timepoint t. The magnitude of each motion has been scaled by their local variance to show all on the same axis.

This means that, when compared to Brownian motion, the case $H < 1/2$ will have stronger local (short-timescale) variations in position with weaker long-range correlations between displacements, while $H > 1/2$ will have weaker local variations with stronger long-range correlations. The cases $H > 1/2$ were the ones originally considered by Hurst [54]. In fact, unless $H = 1/2$, the increments of an FBM are never independent.

Figure B.3 illustrates the process of constructing an FBM by convolving white noise with a Weyl kernel, using a discrete-time approximation. Notice that when

$H < 1/2$, the Weyl kernel accentuates recent noise, and the magnitude of their influence on the present position decays in time. When $H > 1/2$, recent noise has very little influence on the present position, but the influence grows in time. Only when $H = 1/2$ exactly does each noise event have exactly the same effect on the current position, independent of time. As a result, FBM is an appropriate model when the underlying sources of noise that produce changes in position are distributed rather than instantaneous in time. In particular, subdiffusive FBM has found application in viscoelastic dynamics since the sources of noise decay in time.

As the reader may notice, the constructions of Levy and Mandelbrot/Van Ness highlight the analytical properties of FBM, while the Gaussian process construction considered in the opening paragraphs of section B.3 highlights its utility for practical inference. We prefer the Gaussian process construction and the only reason we highlight the analytical constructions, apart from providing historical background on FBM's origin, is that B.7 provides a great deal of intuition about the nature of FBM.

B.6 Simulation of FBM and other Gaussian processes

In practice, while the convolution-based construction illustrated in B.3 is straightforward and provides intuition about the nature of FBM, it does not produce very accurate approximations of FBM.

Instead, an approach based on the Cholesky factorization of the covariance matrix at a discrete set of time points (algorithm B.1) works best. (This is just the regular way to simulate multivariate normal random vectors with arbitrary mean and covariance.) Notice that the Cholesky factor $\mathbf{L}$, which is lower triangular, plays the same role as the Weyl kernel when operating on the white Gaussian noise vector $\mathbf{z}$.
Algorithm **??** is the basis for all of the FBM simulations considered in this thesis.

Algorithm B.1: Simulation of an arbitrary Gaussian process at a discrete set of time points

Parameters:

- The mean function $\mu(t)$ and covariance function $\text{Cov}(t, s)$ associated with the process
- A vector of time points $\mathbf{t} = (t_1, ..., t_n)$

Algorithm:

1. Evaluate the mean $\mu_{\mathbf{X}} = \mu(\mathbf{t})$.
2. Evaluate the covariance matrix $\Sigma = \text{Cov}(\mathbf{t}, \mathbf{t})$.
3. Find the Cholesky decomposition $\Sigma = \mathbf{L}\mathbf{L}^*$, where $\mathbf{L}$ is lower triangular
4. Simulate a standard Gaussian vector $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I)$ of size n
5. Form the product $\mathbf{X} = \mathbf{L}\mathbf{z} + \mu_{\mathbf{X}}$, which is a sample from the Gaussian process.

Appendix C

Appendix: Characteristic functions

In probability, characteristic functions (CFs) provide an alternative to PDFs or CDFs for deriving results on random variables. Getting these results is often simpler when working with the PDF. So we rely on the CF heavily in this thesis.

Here, we review some of the properties of the CF relevant elsewhere in the thesis. Some of these, such as the Fourier slice theorem and the Radon transform, are not usually presented in the context of the CF.

As outlined in the Definitions chapter, we will stick to the following nomenclature:

- $f_X(x)$: probability density function (PDF) for a random variable X
- $F_X(x)$: cumulative distribution function (CDF) for a random variable X
- $\phi_X(x)$: characteristic function (CF) for a random variable X

C.1 Definition

The characteristic function for a random variable X is defined

$$\phi_X(k) = \mathbb{E}\left[e^{ikX}\right] \tag{C.1}$$

The characteristic function of a random variable always exists, even when its PDF does not.

As equation C.1 suggests, if a random variable has a PDF, then its characteristic function is the Fourier transform of its PDF:

$$\mathbb{E}\left[e^{ikX}\right] = \int_{-\infty}^{\infty} e^{ikX} f_X(x)\mathrm{d}x = \mathcal{F}\left[f_X\right](k)$$

Notice in particular that the sign of the exponent is reversed relative to the most common FT definition. For consistency, we have adopted this reversed sign throughout this thesis.

If $\mathbf{X}$ is a random vector, then its CF is defined

$$\phi_{\mathbf{X}}(\mathbf{k}) = \mathbb{E}\left[e^{i\mathbf{k}^T\mathbf{X}}\right]$$

C.2 Moments

Given the exponential series

$$e^x = 1 + x + \frac{x^2}{2} + ... = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

we have

$$\begin{aligned}\phi_X(x) = \mathbb{E}\left[\sum_{n=0}^{\infty} \frac{(ikX)^n}{n!}\right] &= \sum_{k=0}^{\infty} \frac{(ik)^n}{n!}\mathbb{E}\left[X^n\right] \\ &= 1 + ik\mathbb{E}\left[X\right] + \frac{(ik)^2}{2}\mathbb{E}\left[X^2\right] + ...\end{aligned}$$

From this, we immediately acquire the moment-generating formula

$$\mathbb{E}\left[X^n\right] = \frac{1}{i^n}\frac{\partial^n \phi_X(k)}{\partial k^n}\Big|_{k=0} \tag{C.2}$$

If the n^{th} moment of X is defined, then the CF is at least n times differentiable. Notice how the 0^{th} moment of a random variable is always 1, due to the normalization condition on the probability density.

To see how this extends to the multivariate case, we'll look at the situation with two random variables X and Y:

$$\begin{aligned}\phi_{X,Y}(k_x, k_y) = \mathbb{E}\left[1 + i(k_x X + k_y Y) + \frac{i^2}{2}(k_x X + k_y Y)^2 + ...\right] \\ = 1 + ik_x\mathbb{E}\left[X\right] + ik_y\mathbb{E}\left[Y\right] + \frac{i^2}{2}k_x^2\mathbb{E}\left[X^2\right] \\ + \frac{i^2}{2}k_y^2\mathbb{E}\left[Y^2\right] + i^2 k_x k_y \mathbb{E}\left[XY\right] + ...\end{aligned}$$

Then

$$\frac{\partial^n \phi_{X,Y}}{\partial k_x^n} = i^n\mathbb{E}\left[X^n\right] + \left(\text{dependence on } k_x \text{ or } k_y\right)$$

so that

$$\mathbb{E}\left[X^n\right] = \frac{1}{i^n}\frac{\partial^n \phi_{X,Y}}{\partial k_x^n}\Big|_{\mathbf{k}=\mathbf{0}} \tag{C.3}$$

Using a similar logic, we can get the cross moment formula

$$\mathbb{E}\left[X^n Y^m\right] = \frac{1}{i^{n+m}}\frac{\partial^{n+m} \phi_{X,Y}}{\partial k_x^n \partial k_y^n}\Big|_{\mathbf{k}=\mathbf{0}} \tag{C.4}$$

and this easily extends to higher dimensions.

From C.3, we also have

$$\frac{1}{i^2}\nabla^2 \phi_{X,Y}(k_x, k_y)\Big|_{\mathbf{k}=\mathbf{0}} = \mathbb{E}\left[X^2 + Y^2\right] \tag{C.5}$$

This useful, for example, when determining the mean-squared displacement of particles under various types of random motion.

C.3 Sums of independent random variables

Suppose that X and Y are two independent random variables with the PDFs $f_X(x)$ and $f_Y(y)$. We seek the distribution of their sum $X + Y$.

We'll call this sum Z, and we'll call the corresponding PDF $f_Z(z)$. If we fix X at some value x', then the probability that Z takes on the value z is just the probability that Y takes on the value $z - x'$. That is,

$$\Pr\left(Z = z \mid X = x'\right) = \Pr\left(Y = z - x'\right)$$

Then $\Pr\left(Z = z\right)$ can be found via the law of total probability by summing over all possible values for x':

$$f_Z(z) = \int f_X(x) f_Y(z - x) \mathrm{d}x$$

This is a convolution. Since the CF is the Fourier transform of the PDF, this means we can apply the Fourier transform's convolution theorem to write the CF of Z as the simple product of the CFs for X and Y:

$$\phi_Z(k) = \phi_X(k)\phi_Y(k)$$

More generally, for independent random variables X_1, ..., X_n, the CF of their sum is the product of their individual CFs:

$$\phi_{X_1+\ldots+X_n}(k) = \prod_{j=1}^{n} \phi_{X_j}(k) \tag{C.6}$$

The extension to the multivariate case is straightforward. Equation C.6 is endlessly useful for us in this thesis.

C.4 Slice theorem

The characteristic function inherits many of the the powerful projection techniques of the Fourier transform familiar from image processing. This is useful for marginal distributions, since marginalization can be understood as the projection of probability densities. Here, we review the Fourier slice theorem (sometimes known as the "central slice theorem") and apply it to derive marginal distributions of random variables.

What is a marginal distribution? If X and Y are two random variables with the joint PDF $f_{X,Y}(x, y)$, then the marginal distribution of X is obtained by integrating out the dependence on y:

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, y)\mathrm{d}y$$

Intuitively, the marginal distribution represents what we know about X when we lack any knowledge whatsoever about Y. Geometrically, it is equivalent to the projection of the joint PDF onto the x axis.

Substituting the joint characteristic function for $f_{X,Y}$ in the equation above, we have

$$f_X(x) = \int_{-\infty}^{\infty} \mathrm{d}y \left(\frac{1}{(2\pi)^2} \iint_{-\infty}^{\infty} \mathrm{d}k_x \mathrm{d}k_y \, \phi_{X,Y}(k_x, k_y) e^{-i(k_x x + k_y y)} \right)$$

Swapping the order of integration,

$$f_X(x) = \frac{1}{(2\pi)^2} \iint_{-\infty}^{\infty} \mathrm{d}k_x \mathrm{d}k_y \, \phi_{X,Y}(k_x, k_y) e^{-ik_x x} \int_{-\infty}^{\infty} \mathrm{d}y e^{-ik_y y}$$

But we know from the sifting property of the delta function that

$$\int_{-\infty}^{\infty} \delta(k_y) e^{ik_y y} \, \mathrm{d}k_y = 1$$

so that, inverting the transform,

$$\delta(k_y) = \frac{1}{2\pi} \int_{-\infty}^{\infty} (1) e^{-ik_y y} \mathrm{d}y$$

Substituting this into our marginalization equation, we have

$$f_X(x) = \frac{1}{2\pi} \iint_{-\infty}^{\infty} \mathrm{d}k_x \mathrm{d}k_y \, \phi_{X,Y}(k_x, k_y) e^{-ik_x x} \delta(k_y)$$

Again applying the sifting property,

$$\begin{aligned} f_X(x) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \phi_{X,Y}(k_x, 0) e^{-ik_x x} \, \mathrm{d}k_x \\ &= \mathcal{F}^{-1} \left[\phi_{X,Y}(k_x, 0) \right] \end{aligned} \tag{C.7}$$

It is easy to see how this proof extends to higher dimensions. If we have a joint distribution of X_1, ..., X_n with $1 < m < n$ and we wish to marginalize out X_{m+1} through X_n, then we have

$$f_{X_1,\ldots,X_m}(\mathbf{x}) = \frac{1}{(2\pi)^m} \int_{\mathbb{R}^m} \phi_{X_1,\ldots,X_n}(k_1, \ldots, k_m, 0, \ldots, 0) e^{-i(k_1 x_1 + \ldots + k_m x_m)} \, \mathrm{d}\mathbf{k} \tag{C.8}$$

C.5 Radially symmetric densities

Imagine we have some random vector $\mathbf{R} = (R_1, ..., R_n)^T$. To any value of $\mathbf{R}$, we can assign a Euclidean distance from the origin R. We'll write this as

$$R = |\mathbf{R}| = \left(R_1^2 + \ldots + R_n^2 \right)^{1/2}$$

The density function $f_{\mathbf{R}}(\mathbf{r})$ is radially symmetric if it can be expressed as a function of R alone. We usually write this form as $f_{\mathbf{R}}(r)$.

If $f_{\mathbf{R}}(\mathbf{r})$ is radially symmetric, then its characteristic function $\phi_{\mathbf{R}}(\mathbf{k})$ is also radially symmetric. It can be expressed solely as a function of $k = |\mathbf{k}|$, the radial distance from the origin of the Fourier domain.

If $f_{\mathbf{R}}(r)$ is radially symmetric and $\mathbf{R} \in \mathbb{R}^n$, then its characteristic function can be written as the hyperspherical Fourier-Bessel transform

$$\phi_{\mathbf{R}}(k) = \frac{(2\pi)^{\frac{n}{2}}}{k^{\frac{n-1}{2}}} \int_0^{+\infty} r^{\frac{n-1}{2}} f_{\mathbf{R}}(r) J_{\frac{n}{2}-1}(kr) \, (kr)^{\frac{1}{2}} \, \mathrm{d}r \tag{C.9}$$

where $J_{\frac{n}{2}-1}$ is a Bessel function of the first kind of order $\frac{n}{2} - 1$. The inverse is identical except for scaling factors:

$$f_R(r) = \frac{1}{(2\pi)^{\frac{n}{2}} r^{\frac{n-1}{2}}} \int_0^{+\infty} k^{\frac{n-1}{2}} \phi_{\mathbf{R}}(k) J_{\frac{n}{2}-1}(kr) \, (kr)^{\frac{1}{2}} \, \mathrm{d}k$$

The scaling factors are consequences of our definition of the Fourier transform, derived from the CF. It's possible to define the Fourier transform such that equation C.9 becomes truly identical to its inverse, which is the more familiar situation. We're generally not too concerned with scaling factors, since normalization takes care of most of the difficulties for us.

The term

$$\int_0^{+\infty} f_{\mathbf{R}}(r) J_{\frac{n}{2}-1}(kr) r \mathrm{d}r$$

corresponds to a Hankel transform of order $\frac{n}{2}-1$, sometimes written $H_{\frac{n}{2}-1}\left[f_{\mathbf{R}}\right](k)$. As a result, the CF can be expressed

$$\phi_R(k) = \frac{(2\pi)^{n/2}}{k^{\frac{n}{2}-1}} H_{\frac{n}{2}-1}\left[r^{\frac{n}{2}-1} f_{\mathbf{R}}(r)\right](k) \tag{C.10}$$

The Hankel transform is associated with a large number of identities that prove useful when dealing with radially symmetric characteristic functions. An extremely useful reference for these identities is Harry Bateman's book [70], which can be compared directly to equation C.9 to obtain transforms for a variety of radially symmetric functions.

Two of the most useful identities for this thesis are presented here.

(*Identity 1*) If $\phi_{\mathbf{R}}(r) = e^{-ar}$, then

$$f_{\mathbf{R}}(r) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\pi^{\frac{n+1}{2}} a^n \left(1 + \left(\frac{r}{a}\right)^2\right)^{\frac{n+1}{2}}} \tag{C.11}$$

Equation C.11 is useful when dealing with *n*-dimensional Cauchy flights.

(*Identity 2*) If $\phi_{\mathbf{R}}(r) = e^{-ar^2}$, then

$$f_{\mathbf{R}}(r) = \frac{\exp\left(-\frac{r^2}{4a}\right)}{(4\pi a)^{\frac{n}{2}}} \tag{C.12}$$

C.6 Abel and Radon transforms

For practical image processing, the Fourier slice theorem C.8 is often phrased in so-called "cycles of operators". These representations are particularly popular in tomographic reconstruction, but we'll see that they're also useful when working

with probability densities.

The central idea of these "cycles" is the following. Suppose $\mathcal{F}_n$ is the n-dimensional Fourier transform and $\mathcal{P}_{n\to m}$ is some projection operator that projects a function from a space of dimension n onto some subspace of dimension m. Further, let $\mathcal{S}_{n\to m}$ be a "slice operator" that evaluates one or more of the arguments of a function at zero. A few examples:

$$\begin{aligned}\mathcal{S}_{2\to 1}\left[f(x,y)\right] &= f(x,0)\\ \mathcal{S}_{3\to 2}\left[f(x,y,z)\right] &= f(x,y,0)\\ \mathcal{S}_{3\to 1}\left[f(x,y,z)\right] &= f(x,0,0)\end{aligned}$$

and so on. There are multiple choices for a given $\mathcal{S}_{n\to m}$ that will produce different results when the function f is not radially symmetric.

Then one way to state the slice theorem C.8 is

$$\mathcal{F}_m\mathcal{P}_{n\to m}f = \mathcal{S}_{n\to m}\mathcal{F}_n f$$

Exactly what kind of projection is produced depends on the identity of the projection and slice operators. When f is not radially symmetric, then the choice of $\mathcal{S}$ must match the choice of $\mathcal{P}$.

The Abel and Radon transforms represent two specific types of projection operators:

- The Abel transform projects a function out of $\mathbb{R}^n$ into $\mathbb{R}^{n-1}$.
- The Radon transform projects a function out of $\mathbb{R}^n$ into $\mathbb{R}^1$. That is, the function is projected onto a line.

The Abel transform is defined only for radially symmetric functions, which means that the specific identity of $\mathcal{S}_{n\to(n-1)}$ is irrelevant. The Radon transform is also defined for non-radially symmetric functions, but we will generally apply it to symmetric functions and won't dwell too much on the asymmetric cases.

These kind of operators have special significance for the problem of spaSPT state estimation because they represent two extremely important operations in our imaging geometry:

- The Abel transform projects a particle's motion from its native three dimensions onto our camera's two dimensions.
- The Radon transform projects a particle's motion from its native three dimensions onto the one-dimensional axis of our camera. This allows us to directly treat the defocalization problem using density functions defined in 3D.

Motivated by these applications, we focus our attention on the application of these two operators to probability density functions. A more complete treatment can be found in Stanley Deans' chapter [71], from which many of the details in this section are taken.

C.6.1 Abel transforms

The Abel transform of a function f is defined

$$\mathcal{A}[f](y) = 2\int_{|y|}^{\infty} \frac{f(r)r}{\sqrt{r^2 - y^2}} \mathrm{d}r \tag{C.13}$$

Intuitively, the Abel transform is the projection of an n-dimensional function onto an $(n-1)$-dimensional plane. To see this, imagine we have a bivariate function $f_{X,Y}(x,y)$. This function is radially symmetric, so that it can be expressed as some $f_{X,Y}(r)$ with $r = \sqrt{x^2 + y^2}$.

To project this function onto the y-axis, we can take

$$f_Y(y) = \int_{-\infty}^{\infty} f_{X,Y}(r)\,\mathrm{d}x$$

Since $f_{X,Y}(r)$ is even in x, this can also be written

$$f_Y(y) = 2\int_{0}^{\infty} f_{X,Y}(r)\,\mathrm{d}x$$

Now, because $x = \sqrt{r^2 - y^2}$, we have $\mathrm{d}x = r\mathrm{d}r/\sqrt{r^2 - y^2}$. Changing from x to r, the integration limits now run from $r = |y|$ to $r = \infty$, and we have

$$f_Y(y) = 2\int_{|y|}^{\infty} \frac{f_{X,Y}(r)}{\sqrt{r^2 - y^2}} r\mathrm{d}r$$

which brings us back to the definition of the Abel transform. The n-dimensional case follows by replacing our $r = \sqrt{x^2 + y^2}$ with $r = \sqrt{x^2 + y_1^2 + ... + y_{n-1}^2}$ and realizing that none of the above changes.

Now, the slice theorem tells us that this projection can also be represented as the one-dimensional inverse Fourier transform

$$f_Y(y) = \mathcal{F}_1^{-1}\left[\phi_{X,Y}(0,k_y)\right](y)$$

In this way, we have the identification

$$\mathcal{F}_1^{-1}\left[\mathcal{F}_2\left[f_{X,Y}\right](0,k_y)\right](y) = \mathcal{A}\left[f_{X,Y}\right](y)$$

where $\mathcal{F}_1$ and $\mathcal{F}_2$ represent one- and two-dimensional Fourier transforms respectively. This can be simplified further because $f_{X,Y}(x,y)$ is radially symmetric. This means we can apply equation C.10, which says that the two-dimensional Fourier transform of a symmetric function becomes a Hankel transform of order 0. Then

$$H_0\left[f_{X,Y}(r)\right](y) = \mathcal{F}_1\left[\mathcal{A}\left[f_{X,Y}(r)\right]\right](y) \tag{C.14}$$

This is the so-called *FHA* cycle of operators. The letters mean *Fourier-Hankel-Abel*. This extends easily to higher dimensions. Let r_n be the distance of a random vector $\mathbf{R}$ from the origin in $\mathbb{R}^n$ and let $r_{n-1} = \sqrt{r_n^2 - y^2}$ be its distance from the origin in some $(n-1)$-dimensional subspace. y is the component of $\mathbf{R}$ in the direction orthogonal to this subspace. Then

$$\mathcal{F}_n\left[f_{\mathbf{R}}\right](r_{n-1}) = \mathcal{F}_{n-1}\left[\mathcal{A}\left[f_{\mathbf{R}}\right]\right](r_{n-1}) \tag{C.15}$$

Of course, we can drop r_n and r_{n-1} to make this cleaner, but we include them here to illustrate the relationship between the spaces and to show the implicit use of the slice operator in this equation. Both sides of equation C.15 project the function out of an n-dimensional space onto an $(n-1)$-dimensional hyperplane.

Taking advantage of equation C.10, we can also write this in terms of Hankel transforms:

$$\mathcal{H}_{\frac{n}{2}-1}\left[r^{\frac{n}{2}-1}f_R(r)\right] = \mathcal{H}_{\frac{n-3}{2}}\left[r^{\frac{n-3}{2}}\mathcal{A}\left[f_R\right](r)\right]$$

However, we prefer C.15 for clarity.

C.6.2 Radon transforms

The Radon transform of a function $f_{\mathbf{X}}(\mathbf{x})$ in n dimensions is defined

$$\mathcal{R}\left[f_{\mathbf{X}}\right](p,\xi) = \int_{\mathbb{R}^n} f_{\mathbf{X}}(\mathbf{x})\delta\left(p - \xi^T\mathbf{x}\right)\mathrm{d}\mathbf{x} \tag{C.16}$$

where δ is a delta function, and $\xi, \mathbf{x} \in \mathbb{R}^n$.

Since $p - \xi^T\mathbf{x} = 0$ defines a $(n-1)$-dimensional plane in $\mathbb{R}^n$ with the normal vector ξ, the Radon transform can be understood as the projection of an n-dimensional density onto the line $f(p) = p\xi$. If the density $f_{\mathbf{X}}(\mathbf{x})$ is radially symmetric, then the choice of ξ is irrelevant.

It's easy to see that the Abel transform coincides with the Radon transform when working with a radially symmetric function in two dimensions. In three dimensions, the Radon transform is equivalent to two sequential applications of the Abel transform, which we write as $\mathcal{R} = \mathcal{A}^2$, and so on.

Due to the slice theorem C.8, we know that we can also obtain the projection of an n-dimensional density onto a one-dimensional line $\xi \in \mathbb{R}^n$ by taking

$$f_X(x) = \mathcal{F}_1^{-1}\left[\mathcal{F}_n\left[f_{\mathbf{X}}(\mathbf{x})\right](x\xi)\right]$$

Here, $\mathcal{F}_1$ and $\mathcal{F}_n$ represent one-dimensional and n-dimensional Fourier transforms respectively. The term $x\xi = x\left(\xi_1, ..., \xi_n\right)^T$ is a vector in the direction of the line onto which we wish to project. The transform $\mathcal{F}_1^{-1}$ is taken with respect to this line. From a probability perspective, the resulting random variable X on the left side of this equation is defined by a linear combination of the random vector $\mathbf{X}$: $X = \xi_1 X_1 + ... + \xi_n X_n$.

The previous equation leads us to the identification

$$\mathcal{R}\left[f\right](p, \xi) = \mathcal{F}_1^{-1}\left[\mathcal{F}_n\left[f_{\mathbf{X}}(\mathbf{x})\right](p\xi)\right]$$

or, more often,

$$\mathcal{F}_1\left[\mathcal{R}\left[f\right]\right] = \mathcal{F}_n\left[f\right] \tag{C.17}$$

Equation C.17 is a special case of the Fourier slice theorem, with the Radon transform defining the projection operator. It is endlessly useful in applications, particularly in tomographic reconstruction and, for us, when projecting probability densities from 3D to 1D.

C.6.3 Note on efficiency

Not all of the operators $\mathcal{F}_1$, $\mathcal{F}_n$, $\mathcal{A}$, $\mathcal{H}_\nu$, and $\mathcal{R}$ are equally easy for a computer to perform. One of the most important uses of equations like C.15 and C.17 is to produce fast shortcuts for calculations.

For example, Hankel transforms might make math easier in (hyper)spherically symmetric systems, but they are rather difficult to perform numerically. Even the most efficient approaches, such as the method of Ogata 2005 [72], often encounter difficulties due to the highly oscillatory nature of the transform and the Gibbs phenomena at the origin. If we have the radial density of a two-dimensional function, then using the right side of equation C.14 is far preferable to using the left side.

Likewise, in spaces of high dimensionality, taking the n dimensional Fourier transform becomes unfeasible. Indeed, $n = 3$ is impractical for any task that needs to be done quickly, such as curve fitting. In many cases, we are not even interested in the n-dimensional density, and only care about the radial distance from the origin. In these cases, equation C.17 gives us a fast shortcut for computing radial densities. This is heavily exploited in our algorithms for Levy flights.

References

[1] Heim R, Cubitt A, Tsien R (1995). Improved green fluorescence. *Nature* **373** (6516): 663-4 (doi:10.1038/373663b0).

[2] Fick A. Ueber diffusion. *Annalen der Physik* **94**, 59-86 (1855). doi:10.1002/andp.18551700105.

[3] Betzig E. et al. Imaging intracellular fluorescent proteins at nanometer resolution. *Science* **313**, 1642-1645 (2006). doi:10.1126/science.1127344

[4] Rust, M. J., Bates M. & Zhaung X. Stochastic optical reconstruction microscopy (STORM) provides sub-diffraction-limit image resolution. *Nature Methods* **3**, 793-795 (2006).

[5] Barak L. S. & Webb W. W. Diffusion of low density lipoprotein-receptor complex on human fibroblasts. J Cell Biol 95, 846-52 (1982).

[6] Gross D. & Webb W. W. Molecular counting of low-density lipoprotein particles as individuals and small clusters on cell surfaces. *Biophys J* **49**, 901-11 (1986).

[7] Dahan M. et al. Diffusion dynamics of glycine receptors revealed by single quantum dot tracking. *Science* **302**, 442 (2003).

[8] Manley S. et al. High-density mapping of single-molecule trajectories with photoactivated localization microscopy. *Nature Methods* **5**, 155-157 (2008).

[9] English B. P. et al. Single-molecule investigations of the stringent response machinery in living bacterial cells. *Proc Natl Acad Sci* **108**, 12573-12574 (2011).

[10] Izeddin I. et al. Single-molecule tracking in live cells reveals distinct target-search strategies of transcription factors in the nucleus. *Elife* (2014). doi:10.7554/elife.02230

[11] Normanno D. et al. Probing the target search of DNA-binding proteins in mammalian cells using TetR as a model searcher. *Nature Communications* **6**, 7357 (2015).

[12] Los G. V. et al. HaloTag: a novel protein labeling technology for cell imaging and protein analysis. *ACS Chemical Biology* **3(6)**, 373-382 (2008). https://doi.org/10.1021/cb800025k

[13] Tokunaga M. Highly inclined thin illumination enables clear single-molecule imaging in cells. *Nature Methods* **5**, 159-161 (2008). doi:10.1038/nmeth1171

[14] Grimm J. B. et al. A general method to improve fluorophores for live-cell and single-molecule microscopy. *Nature Methods* **12**, 244-250 (2015). doi:10.1038/nmeth.3256

[15] Grimm J. B. et al. Bright photoactivatable fluorophores for single molecule imaging. *Nature Methods* **13**, 985-988 (2016). doi:10.1038/nmeth.4034

[16] Axelrod D. et al. Mobility measurement by analysis of fluorescence photobleaching recovery kinetics. *Biophysical Journal* **16**, 1055-69 (1976). doi:10.1016/S0006-3495(76)85755-4

[17] Magde D., Elson E. & Webb W. W. Thermodynamic fluctuations in a reacting system - measurement by fluorescence correlation spectroscopy. *Physical Review Letters* **29**, 705-708 (1972). https://doi.org/10.1103/PhysRevLett.29.705

[18] Wiedenmann J. *et al.* (2004). EosFP, a fluorescent marker protein with UV-inducible green-to-red fluorescence conversion. *Proc. Natl. Acad. Sci.* **101(45)**, 15905-10 (doi:10.1073/pnas.0403668101),

[19] Xiang L. et al. Single-molecule displacement mapping unveils nanoscale heterogeneities in intracellular diffusivity. *Nature Methods* **17**, 524-530 (2020). doi:10.1038/s41592-020-0793-0

[20] Holcman D. et al. Single particle trajectories reveal active endoplasmic reticulum luminal flow. *Nature Cell Biology* **20**, 1118-1125 (2018). https://doi.org/10.1038/s41556-018-0192-2

[21] Elf J. et al. Probing transcription factor dynamics at the single-molecule level in a living cell. *Science* **316**, 1191-1194 (2007). doi:10.1126/science.1141967

[22] Clarke D. T. & Martin-Fernandez M. L. A brief history of single-particle tracking of the epidermal growth factor receptor. *Methods and Protocols* **2** (2019). doi:10.3390/mps2010012

[23] Hansen A. S. et al. CTCF and cohesin regulate chromatin loop stability with distinct dynamics. *Elife* (2017). doi:10.7554/eLife.25776

[24] McSwiggen D. T. et al. Evidence for DNA-mediated nuclear compartmentalization distinct from phase transition. *Elife* (2019). doi:10.7554/eLife.47098

[25] Alexander J. M. et al. Live-cell imaging reveals enhancer-dependent Sox2 transcription in the absence of enhancer proximity. *Elife* (2019). doi:10.7554/eLife.41769

[26] Bancaud A. et al. Molecular crowding affects diffusion and binding of nuclear proteins in heterochromatin and reveals the fractal organization of chromatin. *EMBO J* **28**, 3785-3798 (2009). doi:10.1038/emboj.2009.340

[27] Shen H. et al. Single particle tracking: from theory to biophysical applications. *ACS Chem Rev* **117**, 7331-7376 (2017). https://doi.org/10.1021/acs.chemrev.6b00815

[28] Berg O. G., Winter R. B. & von Hippel P. H. Diffusion-driven mechanisms of protein translocation on nucleis acids I: models and theory. *Biochemistry* **20**, 6929-6948 (1981). doi:10.1021/bi00527a028

[29] Perrin J. L'agitation moleculaire et le mouvement brownien. *Comptes Rendus Acad Sci* **146**, 967-970 (1908).

[30] Berglund A. J. Statistics of camera-based single-particle tracking. *Phys Rev E* **82**, 011917 (2010). doi:https://doi.org/10.1103/PhysRevE.82.011917

[31] Michalet X. Mean square displacement analysis of single-particle trajectories with localization error: Brownian motion in an isotropic medium. *Phys Rev E Stat Nonlin Soft Matter Phys* **82** (2010). doi:10.1103/PhysRevE.82.041914

[32] Michalet X. & Berglund A. J. Optimal diffusion coefficient estimation in single-particle tracking. *Phys Rev E Stat Nonlin Soft Matter Phys* **85** (2012). doi:10.1103/PhysRevE.85.061916

[33] Abrahamsson S. et al. Fast multicolor 3D imaging using aberration-corrected multifocus microscopy. *Nature Methods* **10**, 60-63 (2013). https://doi.org/10.1038/nmeth.2277

[34] Chenouard N., Smal I., de Chaumont F. et al. Objective comparison of particle tracking methods. *Nat Methods* **11**, 281–289 (2014). https://doi.org/10.1038/nmeth.2808

[35] Yan R. et al. Spectrally resolved and functional super-resolution microscopy via ultrahigh-throughput single molecule spectroscopy. *ACS Chem Res* **51**, 697-705 (2018). https://doi.org/10.1021/acs.accounts.7b00545

[36] Einstein A. On the motion - required by the molecular kinetic theory of heat - of small particles suspended in a stationary liquid. *Annalen der Physik* **17**, 549-560 (1905).

[37] Jain R. & Sebastian K. L. Diffusion in a crowded, rearranging environment. *J Phys Chem B* **120**, 3988-3992 (2016). https://doi.org/10.1021/acs.jpcb.6b01527

[38] Shell S. et al. Dynamic heterogeneity and non-Gaussian behavior in a model supercooled liquid. *J Phys Cond Mat* **17** (2005).

[39] Parthasarathy R. Rapid, accurate particle tracking by calculation of radial symmetry centers. *Nature Methods* **9**, 724-726 (2012). https://doi.org/10.1038/nmeth.2071

[40] Smith C. S. et al. Fast, single-molecule localization that achieves theoretically minimum uncertainty. *Nature Methods* **7**, 373-375 (2010). doi:10.1038/nmeth.1449

[41] Laurence T. A. & Chromy B. A. Efficient maximum likelihood estimator fitting of histograms. *Nature Methods* **7**, 338-339 (2010).

[42] Dill, K. A. & Bromberg, S. *Molecular Driving Forces: Statistical Thermodynamics in Biology, Chemistry, Physics, and Nanoscience.* Garland Science, New York City, 2010 (2 ed).

[43] Crank, J. *The Mathematics of Diffusion.* Clarendon Press, Oxford, 1975.

[44] Bar-Shalom Y., Daum F. & Huang J. The probabilistic data association filter. *IEEE Control Systems* **29**, 82-100 (2009). (doi:10.1109/MCS.2009.934469)

[45] Lee E. H., Zhang Q. & Song T. L. Markov chain realization of joint integrated probabilistic data association. *Sensors* **17**, 2865 (2017). (https://doi.org/10.3390/s17122865)

[46] Nelson E. Dynamical theories of Brownian motion. Princeton University Press, Princeton, 1967. (doi:10.2307/j.ctv15r57jg)

[47] Balcerek M. & Burnecki K. Testing of fractional Brownian motion in a noisy environment. *Chaos, Solitons & Fractals* **140** (2020). (https://doi.org/10.1016/j.chaos.2020.110097)

[48] Bishop C. M. *Pattern Recognition and Machine Learning*. Springer, Berlin, 2006.

[49] Dempster A. P., Laird N. M. & Rubin D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* **39**, 1-38 (1977). https://www.jstor.org/stable/2984875

[50] Persson, F., Lindén, M., Unoson, C. et al. Extracting intracellular diffusive states and transition rates from single-molecule tracking data. *Nat Methods* **10**, 265–269 (2013). https://doi.org/10.1038/nmeth.2367

[51] Mandelbrot B. B. & van Ness J. W. Fractional Brownian motions, fractional noises and applications. *SIAM Review* **10**, 422-437 (1968). (https://doi.org/10.1137/1010093)

[52] Ito K. Stochastic integral. *Proceedings of the Imperial Academy* **20**, 519-524 (1944). (doi:10.3792/pia/1195572786)

[53] Levy P. Random functions: General theory with special reference to Laplacian random functions. *Univ. California Publ. Statist.* **1**, 331-390 (1953).

[54] Hurst H. Long term storage capacity of reservoirs. *Transactions of the American Society of Civil Engineers* **116**, 770-799 (1951).

[55] Matsuoka S, Shibata T & Ueda M. Statistical analysis of lateral diffusion and multistate kinetics in single-molecule imaging. *Biophysical Journal* **97**, 1115-1124 (2009). https://doi.org/10.1016/j.bpj.2009.06.007

[56] Gmachowski L. Fractal model of anomalous diffusion. *Eur Biophys J* **44**, 613-621 (2015). doi:10.1007/s00249-015-1054-5

[57] Ben-Avraham D & Havlin S. *Diffusion and Reactions in Fractals and Disordered Systems*. Cambridge: Cambridge University Press, 2000.

[58] Monnier N. et al. Bayesian approach to MSD-based analysis of particle motion in live cells. *Biophysical Journal* **103**, 616-626 (2012). https://doi.org/10.1016/j.bpj.2012.06.029

[59] Mazza D. et al. A benchmark for chromatin binding measurements in live cells. *Nucleic Acids Research* **40** (2012). https://doi.org/10.1093/nar/gks701

[60] Hansen A.S., Woringer M. et al. Robust model-based analysis of single- particle tracking experiments with Spot-On. *Elife* (2018). doi:10.7554/eLife.33125

[61] Gnedenko B. V. & Kolmogorov A. N. Limit distributions for sums of independent random variables. Cambridge: Addison-Wesley, 1954.

[62] Mandelbrot B. The variation of certain speculative prices. *The Journal of Business* **36**, 394-419 (1963).

[63] Fama E. F. The behavior of stock-market prices. *The Journal of Business* **38**, 34-105 (1965). https://www.jstor.org/stable/2350752

[64] Humphries N. et al. Environmental context explains Levy and Brownian movement patterns of marine predators. *Nature* **465**, 1066-1069 (2010). https://doi.org/10.1038/nature09116

[65] Barrett H. H. The Radon transform and its applications, in *Progress in Optics*. Amsterdam: Elsevier (1984).

[66] Vest C. M. & Steel D. G. Reconstruction of spherically symmetric objects from slit-imaged emission: application to spatially resolved spectroscopy. *Optics Letters* **3**, 54-56 (1978).

[67] Du Mond J. W. M. Compton modified line structure and its relation to the electron theory of solid bodies. *Phys. Rev.* **33**, 643-658 (1929).

[68] Stewart A. T. Momentum distribution of metallic electrons by positron annihilation. *Can. J. Phys.* **35**, 168-183 (1957).

[69] Mijnarends P. E. Determination of anisotropic momentum distribution in positron annihilation. *Phys. Rev.* **160**, 512-519 (1967).

[70] Bateman H. *Tables of Integral Transforms (Volumes I & II)*. New York: McGraw-Hill Book Company (1954).

[71] Deans S. R. Radon and Abel Transforms in *The Transforms and Applications Handbook*. Boca Raton: CRC Press LLC (2000).

[72] Ogata H. A numerical integration formula based on the Bessel functions. *Publications of the Research Institute for Mathematical Sciences* **41**, 949-970 (2005). doi:10.2977/prims/1145474602

[73] Navarro R., Arines J. & Rivera R. Direct and inverse discrete Zernike transform. *Optics Express* **26**, 24269-24281 (2009). https://doi.org/10.1364/OE.17.024269

[74] Genz A. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics* **1**, 141-149 (1992).

[75] Kues T. & Kubitscheck U. Single molecule motion perpendicular to the focal plane of a microscope: application to splicing factor dynamics within the cell nucleus. *Single Molecules* **3**, 218-224 (2002). https://doi.org/10.1002/1438-5171(200208)3:4<218::AID-SIMO218>3.0.CO;2-C

[76] Casella G. & George E. I. Explaining the Gibbs sampler. *The American Statistician* **46**, 167-174 (1992). https://doi.org/10.2307/2685208

[77] Marin J.-M., Mengersen K. & Robert C. P. Bayesian modelling and inference on mixtures of distributions. *Handbook of Statistics* **25**, 459-507 (2005). https://doi.org/10.1016/S0169-7161(05)25016-2

[78] Reisser M., Hettich J., Kuhn T. et al. Inferring quantity and quality of superimposed reaction rates from single molecule survival time distributions. *Scientific Reports* **10**, 1758 (2020). https://doi.org/10.1038/s41598-020-58634-y

[79] Wang B., Kuo J., Bae S. C. & Granick S. When Brownian diffusion is not Gaussian. *Nature Materials* **11**, 481-485 (2012). https://doi.org/10.1038/nmat3308

[80] Lucy L. B. An iterative technique for the rectification of observed distributions. *Astronomical Journal* **79**, 745-754 (1974). doi:10.1086/111605

[81] Kay, S. M. *Fundamentals of Statistical Signal Processing Volume II: Detection Theory*. Prentice Hall PTR, Upper Saddle River, 1998.

[82] Corduneanu A. & Bishop C. M. Variational Bayesian model selection for mixture distributions. *Proceedings of the Eighth International Conference on Artificial Intelligence and Statistics*, 27-34 (2001).

[83] Grosse R. B., Ghahramani X. & Adams R. P. Sandwiching the marginal likelihood using bidirectional Monte Carlo. *arXiv* (2015). arXiv:1511.02543 [stat.ML]

[84] Richardson S. & Green P. J. On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society* **59**, 731-792 (1997). https://doi.org/10.1111/1467-9868.00095

[85] Neal R. M. Bayesian mixture modeling. In: Smith C. R., Erickson G. J. Neudorfer P. O. (eds) Maximum Entropy and Bayesian Methods. *Fundamental Theories of Physics*, vol 50. Springer, Dordrecht (1992).

[86] Blackwell D. Discreteness of Ferguson selections. *Annals of Statistics* **1**, 356-358 (1973). doi:10.1214/aos/1176342373

[87] Blackwell D. & MacQueen J. B. Ferguson distributions via Polya urn schemes. *Annals of Statistics* **1**, 353-355 (1973). doi:10.1214/aos/1176342372

[88] Neal R. M. Markov chain sampling methods for Dirichlet process mixture model. *Journal of Computational and Graphical Statistics* **9**, 249-265 (2000). https://doi.org/10.2307/1390653

[89] Escobar M. D. Estimating normal means with a Dirichlet process prior. *Journal of the American Statistical Association* **89**, 268-277 (1994).

[90] Escobar M. D. & West M. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association* **90**, 577-588 (1995).

[91] Sever R. & Glass C. K. Signaling by nuclear receptors. *Cold Spring Harb Perspect Biol* **5** (2013). doi:10.1101/cshperspect.a016709

[92] Clipsham R. et al. Nr0b1 and its network partners are expressed early in murine embryos prior to steroidogenic axis organogenesis. *Gene Expr Patterns* **4**, 3-14 (2004). doi:10.1016/j.modgep.2003.08.004

[93] Niakan K. et al. Novel role for the orphan nuclear receptor Dax1 in embryogenesis, different from steroidogenesis. *Mol Genet Metab* **88**, 261-271 (2006). doi:10.1016/j.ymgme.2005.12.010

[94] Khalfallah O. et al. Dax-1 knockdown in mouse embryonic stem cells induces loss of pluripotency and multilineage differentiation. *Stem Cells* **27**, 1529-1537 (2009). doi:10.1002/stem.78.

[95] Zanaria E. et al. An unusual member of the nuclear hormone receptor superfamily responsible for X-linked adrenal hypoplasia congenita. *Nature* **372**, 635-641 (1994). https://doi.org/10.1038/372635a0

[96] Matsusue K. et al. Ligand-activated PPARbeta efficiently represses the induction of LXR-dependent promoter activity through competition with RXR. *Mol Cell Endocrinol* **15**, 23-33 (2006). doi:10.1016/j.mce.2006.05.005.

[97] Garcia-Villalba P. et al. Interaction of thyroid hormone and retionic acid receptors on the regulation of the rat growth hormone gene promoter. *Biochem Biophys Res Commun* **15**, 580-586 (1993). doi:10.1006/bbrc.1993.1257

[98] Hunter J. et al. Crosstalk between the thyroid hormone and peroxisome proliferator-activated receptor in regulating peroxisome proliferator-response genes. *Mol Cell Endocrinol* **5**, 213-221 (1996). doi:10.1016/0303-7207(95)03717-9

[99] De Braekeleer E. et al. RARA fusion genes in acute promyelocytic leukemia: a review. *Expert Rev Hematol* **7**, 347-357 (2014). doi:10.1586/17474086.2014.903794

[100] Zhu J. et al. RXR is an essential component of the oncogenic PML/RARA complex in vivo. *Cancer Cell* **12**, 23-25 (2007). https://doi.org/10.1016/j.ccr.2007.06.004

[101] Brazda P. et al. Live-cell fluorescence correlation spectroscopy dissects the role of coregulator exchange and chromatin binding in retinoic acid receptor mobility. *J Cell Sci* **124**, 3631-3642 (2011). doi:10.1242/jcs.086082

[102] Zhu J. et al. Retinoic acid induces proteasome-dependent degradation of retinoic acid receptor α (RARα) and oncogenic RARα fusion proteins. *Proc Natl Acad Sci* **96**, 14807-14812 (1999). doi:10.1073/pnas.96.26.14807

[103] Sucov H. M. et al. Characterization of an autoregulated response element in the mouse retinoic acid receptor type beta gene. *Proc Natl Acad Sci* **87**, 5392-5396 (1990). doi:10.1073/pnas.87.14.5392

[104] Rochette-Egly C. et al. Retinoic acid signaling and mouse embryonic stem cell differentiation: cross talk between genomic and non-genomic effects of RA. *Biochem Biophys Acta* **1851**, 66-75 (2015). doi:10.1016/j.bbalip.2014.04.003

[105] Gianni M. et al. AM580, a stable benzoic acid derivative of retinoic acid, has powerful and selective cyto-differentiating effects on acute promyelocytic leukemia cells. *Blood* **87**, 1520-1531 (1996).

[106] Penvose A. et al. Comprehensive study of nuclear receptor DNA binding provides a revised framework for understanding receptor specificity. *Nature Communications* **10**, 2514 (2019). https://doi.org/10.1038/s41467-019-10264-3

[107] Li J. et al. Both corepressor proteins SMRT and N-CoR exist in large protein complexes containing HDAC3. *EMBO J* **19**, 4342-4350 (2000).

[108] Brazda P. et al. Ligand binding shifts highly mobile retinoid X receptor to the chromatin-bound state in a coactivator-dependent manner, as revealed by single-cell imaging. *Molecular and Cellular Biology* **34**, 1234-1245 (2014). doi:10.1128/MCB.01097-13

[109] Pettitt S. J. et al. Agouti C57BL/6N embryonic stem cells for mouse genetic resources. *Nature Methods* **6**, 493-495 (2009). https://doi.org/10.1038/nmeth.1342

[110] Hoffman L. M. et al. BMP action in skeletogenesis involves attenuation of retinoid signaling. *J Cell Biol* **174**, 101-113 (2006). doi:10.1083/jcb.200604150

[111] Ran F. A. et al. Genome engineering using the CRISPR-Cas9 system. *Nature Protocols* **8**, 2281-2308 (2013). https://doi.org/10.1038/nprot.2013.143

[112] Sergé A., Bertaux N., Rigneault H. & Marguet D. Dynamic multiple-target tracing to probe spatiotemporal cartography of cell membranes. *Nature Methods* **5**, 687-694 (2008).