A jump distance based parameter inference scheme for particulate trajectories in biological settings

Rebecca Menssen[1], Madhav Mani[1,2*]

1 Department of Engineering Sciences and Applied Mathematics, Northwestern University, Evanston, Illinois, United States of America **2** Department of Molecular Biosciences, Northwestern University, Evanston, Illinois, United States of America

* madhav.mani@northwestern.edu

Abstract

Modern biology is a treasure trove of data. With all this data, it is helpful to have analytical tools that are applicable regardless of context. One type of data that needs more quantitative analytical tools is particulate trajectories. This type of data appears in many different contexts and across scales in biology: from inferring statistics of a bacteria performing chemotaxis to the mobility of ms2 spots within nuclei. Presently, most analyses performed on data of this nature has been limited to mean square displacement (MSD) analyses. While simple, MSD analysis has several pitfalls, including difficulty in selecting between competing models, how to handle systems with multiple distinct sub-populations, and parameter extraction from limited time-series data sets. Here, we provide an alternative to MSD analysis using the jump distance distribution (JDD) [1, 2]. The JDD resolves several issues: one can select between competing models of motion, have composite models that allow for multiple populations, and have improved error bounds on parameter estimates when data is limited. A major consequence is that you can perform analyses using a fraction of the data required to get similar results using MSD analyses, thereby giving access to a larger range of temporal dynamics when the underlying stochastic process is not stationary. In this paper, we construct and validate a derivation of the JDD for different transport models, explore the dependence on dimensionality of the process (1-3 dimensions), and implement a parameter estimation and model selection scheme. Finally, we discuss extensions of our scheme and its applications to biological data.

Author summary

Mean square displacement (MSD) analyses have been the standard for analyzing particulate trajectories, where its shortcomings have been overlooked in light of its simplicity. The Jump Distance Distribution (JDD) has been proposed by others in the past as a new way to analyze particulate trajectories, but has not been sufficiently analyzed in varying numbers of dimensions or given a robust analysis on performance and how it compares to MSD analysis. We present the forms of the JDD in 1, 2, and 3 dimensions for three different models for transport: pure diffusion, directed diffusion, and anomalous diffusion. We also discuss how to select between competing models, and verify our method with a rigorous analysis. Through this, we have a method that is superior to a MSD analysis. This method works across a wide range of parameters, which should make it broadly applicable to any system where the underlying motion is stochastic.

Introduction

Particulate trajectories are seen throughout biological data. This type of data spans length and time scales from the chemotaxis of bacteria [3, 4] to the motion of ms2 spots in Drosophila embryos [5]. When analyzing particulate trajectories, on a basic level, one would like to know how the particles are moving (e.g. diffusion), and what parameters (e.g. diffusion constant), guide that motion.

Mean squared displacement

Typically particulate trajectory analysis has been done through the use of the mean squared displacement (MSD). The most basic version of mean square displacement for N particulate trajectories is defined by Eq (1).

$$\mathrm{MSD}(t) = \frac{1}{N}\sum_{i=1}^{N}(x_n(t) - x_n(0))^2 \tag{1}$$

There are more complicated ways to calculate the MSD, such as doing a sweeping average (called a time-averaged MSD), but the general concept remains the same. The MSD is simple to calculate and there are well defined forms that it follows [2, 6, 7] for different modes of transport. Eq (2a) – Eq (2c) give these forms for a purely diffusive system (D), a directed diffusion system (V), and a constrained or anomalous diffusion system (A), based on how we simulated data [8]. These equations hold across dimensions, with only a constant d that changes depending on the dimensionality of the system. Fig 1B shows what the mean square displacement looks like for each model in one dimension.

$$\mathrm{MSD}_D(t) = 2dDt \tag{2a}$$

$$\mathrm{MSD}_V(t) = 2dD_v t + V^2 t^2 \tag{2b}$$

$$\mathrm{MSD}_A(t) = 2dD_\alpha t^\alpha / \Gamma(1+\alpha) \tag{2c}$$

Pitfalls of MSD analysis include the requirement for many data points [9], difficulties in selecting between competing models [10], how to handle systems with two distinct subpopulations [9, 11], and the relatively large errors in cases when data is limited. Several studies have tried to improve and expand upon the MSD, and also have proposed new methods of extracting models and parameters [9–16].

The jump distance distribution

As an alternative to the MSD, we propose the use of the Jump Distance Distribution (JDD) to classify particulate trajectories [1]. The JDD is closely related to the MSD. Each point on the MSD curve is the mean of the underlying JDD, so by using the JDD, we examine a full distribution as opposed to a set of distribution means.

The idea of the JDD and its potential uses for parameter extraction is not a new one, but so far its use has been limited to purely diffusive systems in two dimensions with an assumed number of population sub-fractions [17–21] or has considered multiple models, but also only in two dimensions [1]. Additionally, little work has been done on analyzing the improvement of the JDD on the MSD in anything other than two dimensional pure

diffusion [20]. Complete derivations of theoretical forms are missing and the treatment of directed and anomalous diffusion is lacking. This work serves to complete the fragmented picture of the JDD and serve as an easy template for model selection and parameter extraction.

The JDD is a frequency distribution of Euclidean distances for points separated by a time lag of τ seconds. Creating the JDD can be done by breaking up trajectory data into intervals of length τ, or by using a length τ sliding window to maximize data (see Constructing the JDD for more detailed information). From this, N data points are generated, binned into a histogram with N_b bins. Fig 1C shows JDDs created from simulated data for three different models.

Analogous to MSD analysis, we derive the closed form mathematical solutions for the JDD. These forms are dependent on the mode of transport and the dimensionality of the system. Table 1 lists the closed forms for pure diffusion, directed diffusion, and anomalous diffusion in one, two, and three dimensions and Fig 1C graphically shows what the closed form solution looks like upon a JDD given the simulated parameters (i.e. diffusion constant) of the system, the time lag τ, the bin spacing (dr), and the bin center positions (r_j for bin j). Note that the bin spacing and bin center positions depend upon the number of bins chosen in creating the JDD frequency distribution.

The method presented here can account for two or more distinct subpopulations of motion occurring in the data, or if there is a switch in motion at a certain point, [1, 17–19] by multiplying each type of motion by the fraction undergoing (or fraction of time in) the motion, and adding the distributions together. We will not focus on this in our paper, but the extension is straightforward to implement given our method and does not rely on both populations undergoing the same type of motion.

The paper is organized as follows. In the Methods section we describe the JDD and its closed mathematical forms, how to simulate data and turn trajectory data into the JDD, how to approach parameter fitting, and finally how to select among competing models of motion. This gives us a complete processing pipeline that can be used to analyze particulate trajectories. In the Results section, we discuss our rationale and findings that shaped our parameter fitting scheme, show parameter fitting and model selection results for a broad range of parameters, and provide evidence for this new analysis technique being an improvement on the MSD. Finally, in the Discussion section, we discuss the application of this method to biological data.

Methods

Pipeline for processing data

The proposed pipeline has three major components.

1. Construct JDD
    - Collect particulate trajectory data
    - Choose lag time τ + number of bins $N_b \rightarrow$ construct the JDD
2. Parameter estimation
    - Use MSD to seed parameter fitting scheme + fit each model using non-linear weighted least squares $\rightarrow \hat{\beta}$, the set of Maximum-Likelihood parameters for all models.
    - Bootstrap to define error bounds on parameters $\rightarrow d\hat{\beta}$.

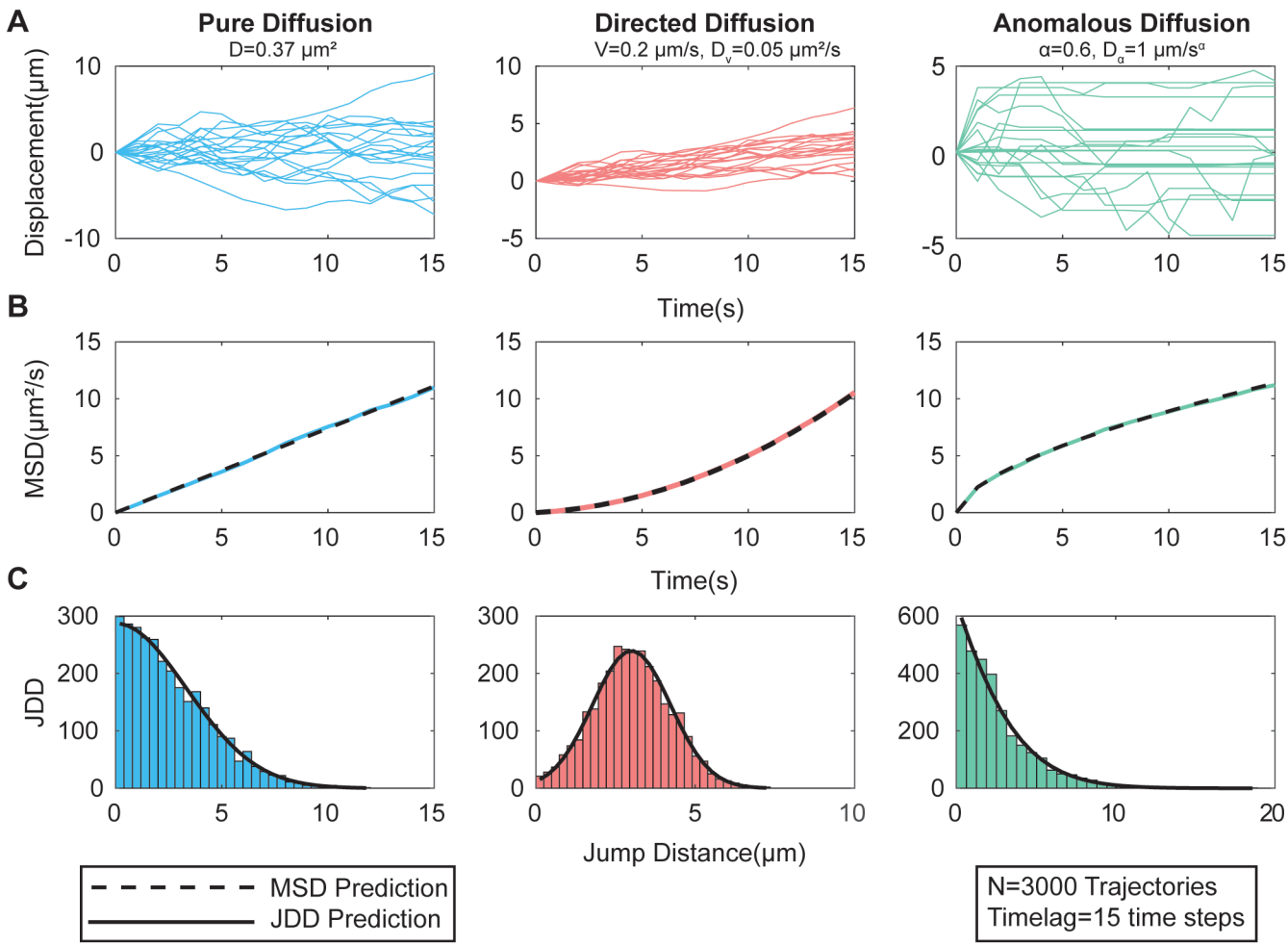

Fig 1. Trajectories, MSDs, and JDDs for three types of diffusion A: Examples of simulated trajectories are used to create MSD and JDD plots. B: Mean squared displacement for each model and the MSD that is predicted using the simulation parameters C: The JDD created from 3000 trajectories and the expected JDD form given the simulated parameters of the system.

3. Model selection

   - Integrate models over the parameter ranges $\hat{\beta} \pm 2d\hat{\beta}$ + Normalize by the length of the integration range (per parameter) $\rightarrow P(JDD|M)$, the probability of observing the data given the model.
   - Employing Bayes Theorem gives $P(M|JDD) \rightarrow$ model selection.

This method is outlined mathematically in Fig 2.

Creation of simulated data and JDD

Simulating particulate trajectories

In this study, we use simulated data and validated our method in one dimension. Pure diffusion was simulated using random Gaussian steps with a variance $2Ddt$ at each time point. Directed diffusion was simulated with a deterministic step of Vdt and a random

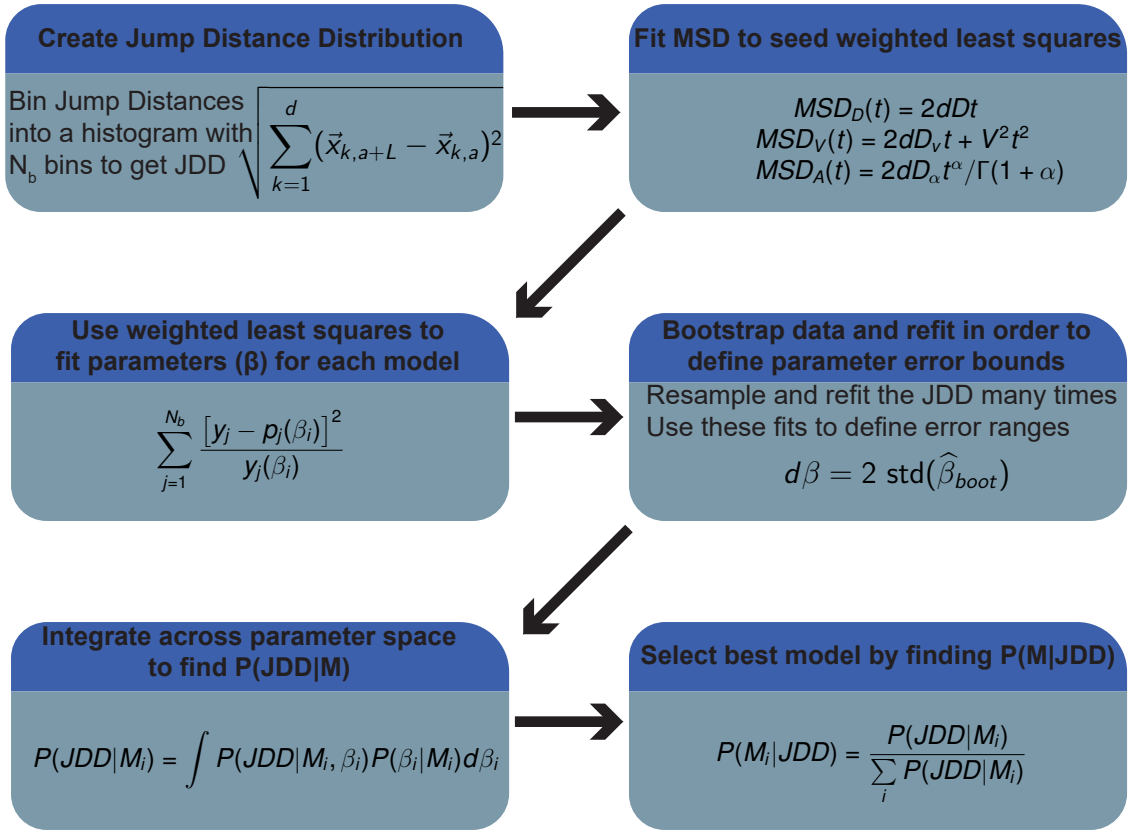

Fig 2. Pipeline for analyzing particulate trajectory data: This figure shows the general method that should be implemented to analyze particulate trajectory data. The result of the pipeline is a set of best fit parameters for each competing model, and a selection of which model was most likely to have created the data. Each step in this figure lists a major equation that is used in that step, but more complete details are given in the methods section and in the code posted on GitHub.

Gaussian step of $2D_V dt$. Anomalous diffusion was simulated using a continuous time random walk (CTRW) [22, 23] using a waiting time as drawn from a generalized Mittag-Leffler function [8] and a random Gaussian steps at each moving point of the size $2D_\alpha dt'^\alpha$, where we set dt' to be the same as the parameter ξ from the Mittag-Leffler function we drew from. With the CTRW, the time of each move does not correspond exactly to a set time step, requiring projection onto a predetermined grid of time steps.

These simulation methods can be extended to two and three dimensions by making the Gaussian steps in each direction, and in the case of direction motion, splitting up the deterministic step into each dimension by the relevant polar and spherical transformations.

Constructing the JDD

Constructing a JDD requires calculating the Euclidean distances between two points on a trajectory a time lag τ apart, and binning them into a histogram with a chosen number of bins, N_b. Insensitivity of the estimated parameters to the choice of time lag is required. S2 Table shows an analysis of the effect of time lag on parameter estimation. Too short or too long of a time lag can have negative effects on parameter estimation or cloud the effects of non-stationary parameters. As a rule of thumb, we initially choose

N_b=$N/100$, as this usually gives a sufficient number of data points for fitting without leaving empty bins. We change the number of bins as needed to improve fitting. In S3 Table we analyze the effect of varying the number of bins on three data sets. Too few bins and the shape of the JDD can change, greatly affecting parameter fitting by changing the skewness of the distribution. If too many bins are used, the bins can become sparsely populated (even empty), and this can have a large effect on parameter fitting accuracy. This is particularly a problem with anomalous diffusion. N_b must be determined keeping these factors in mind.

Beyond the choice of a time lag and number of bins, the other choice in constructing the JDD is how much the data should overlap with itself. The naive choice in constructing the JDD from trajectory is just splitting up the data into independent intervals $\tau/dt + 1$ points long. While this avoids overlapping or correlated data, it requires many trajectories in order to have enough data points. As an alternative, a sliding window JDD can be employed. A sliding window can be constructed by taking a trajectory of length S and splitting it into trajectories of length $\tau/dt + 1$ points long as such, [{1,1+τ/dt},{2,2+τ/dt},{3,3+τ/dt}...{S-τ/dt-1,S}], where the numbers represent the index in a trajectory, allowing the construction of a JDD. A second option is to use this sliding window method, but not use consecutive points, i.e. [{1,1+τ/dt},{3,3+τ/dt},{5,5+τ/dt}...{S-τ/dt-2,S}]. This still gives more data than a long trajectory being cut up, but reduces correlations between data points.

In our initial analysis, we simulated trajectories of length $\tau/dt + 1$, so every JDD data point is independent. This demonstrates the best this method can perform, but other than an academic study, is unlikely to be the method of choice since it requires many trajectories. In S4 Table and S5 Table we compare non-sliding and sliding JDDs for their accuracy in parameter fitting and model selection for pure and directed diffusion.

Parameter estimation and closed form JDDs

Derivation of closed form JDD

Our parameter estimation scheme relies on non-linear weighted least squares estimation. In order to perform this type of estimation, we need to have a closed form solution for each method we are examining. This required us to compile and re-derive prior work on the JDD in two dimensions [1, 17–21], and derive the closed form solutions in one and three dimensions. In the case of pure diffusion, we can solve the relevant diffusion equation [2, 6, 18, 24]. For Directed Diffusion, we were able to perform transformations on the Pure Diffusion closed form solution [2, 6]. Deriving the Anomalous Diffusion form relies on finding the relevant propagator underlying an anomalous system [22]. S1 Appendix gives full derivations for finding the JDD for each method and as a function of dimensionality. These results are compiled in Table 1.

Parameter estimation and estimation error

Given a constructed JDD, parameter estimation can be done through non-linear weighted least squares (NLWLS) – Eq 3. The scheme requires the sample probabilities, y_j, and the closed form expectations, p_j. NLWLS requires an initial guess for parameter values, which were acquired through basic MSD analysis of the data. Additionally, we chose to weight the scheme owing to the heteroskedasticity that is present in the errors of bin counts (See Results-Optimal weighting scheme for our full explanation). We chose a weighting to be the reciprocal of the observed probabilities ($1/y_j$), since our analysis demonstrates that the errors are Poissonian. To account for the possibility of

Table 1. Closed form JDD (frequency distribution) value for bin j for three types of diffusion in one, two, and three dimensions

Method	Dimension	JDD value for bin j
Pure Diffusion	1D	$\frac{Ndr}{\sqrt{\pi D\tau}} \exp\left(\frac{-r_j^2}{4D\tau}\right)$
	2D	$\frac{Ndr\, r_j}{2D\tau} \exp\left(\frac{-r_j^2}{4D\tau}\right)$
	3D	$\frac{Ndr\, r_j^2}{2\sqrt{\pi}(D\tau)^{3/2}} \exp\left(\frac{-r_j^2}{4D\tau}\right)$
Directed Diffusion	1D	$\frac{Ndr}{\sqrt{4\pi D_V\tau}} \exp\left(\frac{-(r_j^2+V^2\tau^2)}{4D_V\tau}\right) \exp\left(\frac{Vr_j}{2D_V}\right)$
	2D	$\frac{Ndr\, r_j}{2D_V\tau} \exp\left(\frac{-(r_j^2+V^2\tau^2)}{4D_V\tau}\right) I_0\left(\frac{Vr_j}{2D_V}\right)$
	3D	$\frac{Ndr\, r_j^2}{2\sqrt{\pi}(D_V\tau)^{3/2}} \exp\left(\frac{-(r_j^2+V^2\tau^2)}{4D_V\tau}\right) \frac{2D_V \sinh\left(\frac{Vr_j}{2D_V}\right)}{Vr_j}$
Anomalous Diffusion	1D	$\frac{Ndr}{2\pi\sqrt{D_\alpha}} \int_{-\gamma+iT}^{\gamma+iT} \exp(ip\tau)\,(ip)^{\alpha/2-1} \exp\left(\frac{-r_j\,(ip)^{\alpha/2}}{\sqrt{D_\alpha}}\right) dp$
	2D	$\frac{Ndr\, r_j}{2\pi D_\alpha} \int_{-\gamma+iT}^{\gamma+iT} \exp(ip\tau)\,(ip)^{\alpha-1} K_0\left(\frac{r_j\,(ip)^{\alpha/2}}{\sqrt{D_\alpha}}\right) dp$
	3D	$\frac{Ndr\, r_j}{2\pi D_\alpha} \int_{-\gamma+iT}^{\gamma+iT} \exp(ip\tau)\,(ip)^{\alpha-1} \exp\left(\frac{-r_j\,(ip)^{\alpha/2}}{\sqrt{D_\alpha}}\right) dp$

empty bins, each bin was given one extra count so that no bin would be weighted with infinite weight.

To perform NLWLS on 3, we used a Levenberg-Marquardt algorithm [25,26]on Eq 3, so in this case β_i is for model i(i being D,V, or A).

$$\sum_{j=1}^{N_b} \frac{[y_j - p_j(\beta_i)]^2}{y_j(\beta_i)} \tag{3}$$

To quantify the error in our parameter fitting scheme, we employed bootstrapping [27]. The standard deviation of the inferred parameters, $d\hat{\beta}$, following bootstrapping of the JDD provides an error bound, which will also be used in the model selection portion of the pipeline.

Model Selection

Bayesian inference scheme

Following the above steps we employ a Bayesian scheme to select the model that best fits the data [1,15]. The Bayesian scheme's prior (Eq 4) assumes that all models (M_i)

are equally probable.

$$P(M_i|JDD) = \frac{P(JDD|M_i)P(M_i)}{\sum_i P(JDD|M_i)P(M_i)} = \frac{P(JDD|M_i)}{\sum_i P(JDD|M_i)} \tag{4}$$

Given the fit parameters and their standard deviations, $\hat{\beta} \pm 2d\hat{\beta}$, we can calculate $P(JDD|M_i)$ with the probability integration scheme seen in Eq 5 [1, 15]. We assume that $P(\beta_i|M_i)$ is uniform based on the range of β and is multiplicative.

Assuming that trajectories are independent, $P(JDD|M_i, \beta_i)$ satisfies a multinomial distribution [1], which then be approximated by Eq 6. Even when trajectories are not independent, this approximation works well.

$$P(JDD|M_i) = \int P(JDD|M_i, \beta_i)P(\beta_i|M_i)d\beta_i \tag{5}$$

$$P(JDD|M_i, \beta_i) = \frac{\sqrt{2\pi N}}{\prod_{j=1}^{N_b} \sqrt{2\pi N p_j(\beta_i)}} \exp\left(-\frac{N}{2}\sum_{j=1}^{N_b} \frac{[y_j - p_j(\beta_i)]^2}{p_j(\beta_i)}\right) \tag{6}$$

After finding $P(JDD, M_i)$ for all possible models, a Bayesian selection scheme, as outlined in Eq 4, can perform model selection.

Results

We subjected the protocol outline in the methods section to the following three tests: 1) Are there benefits to weighted non-linear least squares in the fitting of parameters?, 2) How accurate is the overall method in recovering parameter values and models?, and 3) What are the relative performances of the JDD and MSD based methods?

Optimal Weighting Scheme

Typically, when one uses weights with least squares methods, the weights are proportional to the variance of the data. If the distribution within each bin is Poissonian, then we expect the variance ((Predicted Counts-Actual Counts)^2) to be equal to the mean (Actual Counts).We confirm these with simulations. For each method, we simulated 500 JDDs, and computed the average value of, $[N * y_j - N * p_j(\beta_M)]^2$. This represents the variance in the data, which scales linearly as a function of the average count per bin. Our results for all three transport modes are shown in Fig 3. The distribution is manifestly Poissonian, justifying our weighting scheme.

Bayesian Inference and Parameter Estimation Results

To validate the parameter estimation and Bayesian inference scheme, we performed a broad sweep across parameters and time step to see how errors varied as a function of parameters.

Table 2 shows average results across three different timescales for a variety of parameters. It is important to note that this table was made with JDDs that were constructed without a sliding window. S4 Table and S5 Table show a comparison between non-sliding and sliding data. Non-sliding data gives more accurate results, but if data is limited, the results for using a sliding window still outperform an MSD analysis (see later).

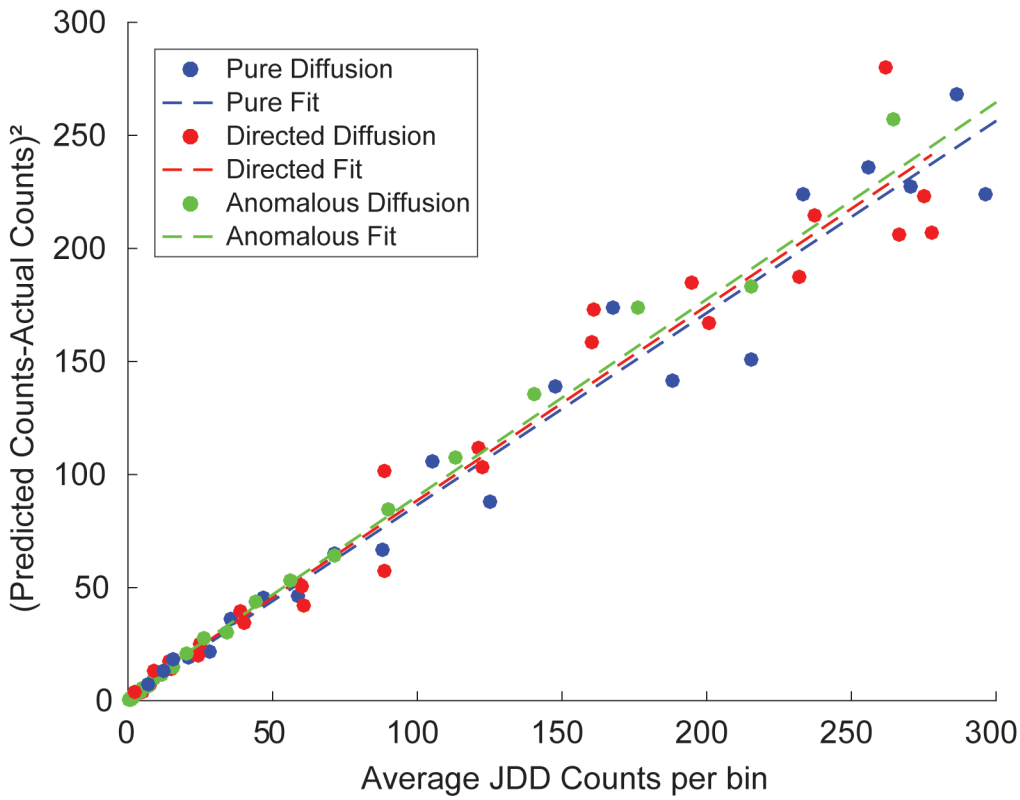

Fig 3. Error Model for the Jump Distance The squared error of the predicted JDD counts and the actual JDD counts compared to the actual JDD counts. The linear relationship between the two suggests that the errors are Poissonian in nature, and thus we should use a weighting of $1/y$, where y is the actual JDD probabilities, to implement in our weighted least squares fitting.

We kept the time lag fixed at $20dt$ for the results presented in this paper, discussions of how a changing time lag affects results are left to the Supplementary Table, S2 Table. We simulated 3000 independent trajectories, and used 30 bins to create our JDD. Similarly, we leave the discussion on the effect of the number of bins on parameter fitting to S3 Table.

Pure Diffusion

Pure Diffusion has the most robust results of the three models. Across the three time steps and diffusion parameters, the average error was less than two percent, and the error bound encapsulated the true simulated parameter. Often, an anomalous model with an exponent close to one is selected in preference to a purely diffusive model, which is a superficial feature of the scheme. Penalizing the number of parameters by making more complicated models less likely or by integrating over a larger parameter range suppresses this effect.

Directed Diffusion

We tested three cases of the relationship between the directed motion parameter (V) and the diffusion constant (D_V): $V \sim D_V$, $V > D_V$, and $D_V > D$, with three different time steps.

In all three cases, we had inaccuracies in both parameter estimation and model selection for a small time step (.1 s) that decreased with increasing its value. This can be understood straightforwardly – inaccuracies are substantial when $V^2\tau^2 \sim D_V\tau$, that is, when the length scale associated with diffusive and ballistic motion balance. An appropriate choice of τ, 5-10 times larger than $\tau \gg D/V^2$, mitigates the above inaccuracies.

Anomalous Diffusion

Given the complexity of the functional forms of anomalous diffusion JDDs, we anticipate that parameter fitting results are less accurate in many cases. These inaccuracy stem from the multiple choices in the analysis: the number of bins, time lag, and the inverse Laplace transform that is part of the closed form JDD. Numerical evaluation of the closed functional form required exploring different integration cutoffs and breaking up the domain of integration, more details of which are explored in S6 Appendix. The methods discussed there are possible ways to improve parameter fitting.

Regardless of the α used, results for fitting α were better than that for D_α, with all errors on $\hat{\alpha}$ below 5% and standard errors below 10 %. Errors for D_α (both in terms of absolute error and standard deviation) were larger as dt increased.

Table 2. Bayesian Selection and Parameter Estimation Results

		Pure Diffusion		**Directed Diffusion**			**Anomalous Diffusion**		
		$D = 0.1\ \mu m^2/s$		$V = 0.5\ \mu m/s$, $D_V = 0.5\ \mu m^2/s$			$\alpha = 0.4$, $D_\alpha = 1\ \mu m^2/s^\alpha$		
		Prob.	$\widehat{\boldsymbol{D}}$	**Prob.**	$\widehat{\boldsymbol{V}}$	$\widehat{\boldsymbol{D}}_{\boldsymbol{V}}$	**Prob.**	$\widehat{\boldsymbol{\alpha}}$	$\widehat{\boldsymbol{D}}_{\boldsymbol{\alpha}}$
	0.1	[57 0 43]	$.0987 \pm .0028$	[30 0 70]	$.5340 \pm .0118$	$.3451 \pm .0114$	[0 0 100]	$.4000 \pm .0094$	$.9880 \pm .0431$
	1	[78 0 22]	$.0985 \pm .0028$	[0 100 0]	$.4922 \pm .0048$	$.5156 \pm .0181$	[0 0 100]	$.4005 \pm .0076$	$.9908 \pm .0497$
	10	[88 0 12]	$.0987 \pm .0028$	[0 100 0]	$.4994 \pm .0014$	$.4935 \pm .0146$	[0 0 100]	$.4007 \pm .0002$	$1.083 \pm .1766$
Time Step (s)		$D = 1\ \mu m^2/s$		$V = 1\ \mu m/s$, $D_V = 0.5\ \mu m^2/s$			$\alpha = 0.6$, $D_\alpha = 1\ \mu m^2/s^\alpha$		
		Prob.	$\widehat{\boldsymbol{D}}$	**Prob.**	$\widehat{\boldsymbol{V}}$	$\widehat{\boldsymbol{D}}_{\boldsymbol{V}}$	**Prob.**	$\widehat{\boldsymbol{\alpha}}$	$\widehat{\boldsymbol{D}}_{\boldsymbol{\alpha}}$
	0.1	[54 0 46]	$.9853 \pm .0282$	[0 8 92]	$.9217 \pm .0159$	$.5219 \pm .0169$	[0 0 100]	$.6262 \pm .0585$	$.9452 \pm .0478$
	1	[79 0 21]	$.9875 \pm .0284$	[0 100 0]	$.9982 \pm .0043$	$.4937 \pm .0148$	[0 0 100]	$.6280 \pm .0492$	$.8750 \pm .1292$
	10	[87 0 13]	$.9853 \pm .0282$	[0 100 0]	$.9998 \pm .0014$	$.4915 \pm .0147$	[0 0 100]	$.6183 \pm .0471$	$.8737 \pm .2220$
		$D = 10\ \mu m^2/s$		$V = 1\ \mu m/s$, $D_V = 1.5\ \mu m^2/s$			$\alpha = 0.8$, $D_\alpha = 1\ \mu m^2/s^\alpha$		
		Prob.	$\widehat{\boldsymbol{D}}$	**Prob.**	$\widehat{\boldsymbol{V}}$	$\widehat{\boldsymbol{D}}_{\boldsymbol{V}}$	**Prob.**	$\widehat{\boldsymbol{\alpha}}$	$\widehat{\boldsymbol{D}}_{\boldsymbol{\alpha}}$
	0.1	[57 0 43]	$9.875 \pm .2836$	[7 0 93]	$1.001 \pm .0211$	$1.138 \pm .0359$	[0 0 100]	$.8192 \pm .0360$	$.9643 \pm .0368$
	1	[78 0 22]	$9.853 \pm .2814$	[0 100 0]	$.9923 \pm .0078$	$1.504 \pm .0490$	[0 0 100]	$.8189 \pm .0355$	$.9317 \pm .0948$
	10	[88 0 12]	$9.875 \pm .2836$	[0 100 0]	$.9993 \pm .0024$	$1.478 \pm .0425$	[0 0 100]	$.8159 \pm .0358$	$.8949 \pm .1633$

Table Notes: The probability for each model (given in the order pure, directed, anomalous) is given for a set of parameters and a set time step. For each combination of parameters on the chart, we simulated 50 sets of trajectories, and took the median parameter value. For the error bound, for each set, we bootstrapped the JDD 50 times and then found the standard deviation in the bootstrapped set fit parameter values, then taking the median over the 50 standard deviations to get the error bound. Only the constants relevant to the model we simulated are given, even if there might have been a better model.

JDD as an improvement on MSD

We demonstrate that the JDD-based method developed here is a significant improvement compared to an MSD analysis, both in terms of accuracy and error bounds.

Explicit simulations showing the JDD outperforming MSD in the data-poor limit were performed, by selecting a number of trajectories, a time lag, and trajectory length and performing JDD and MSD analysis using a sliding window to construct the JDD and MSD. The results of these simulations are displayed in Fig 4 comparing the performance of the two methods for the cases of pure and directed diffusion.

Summarizing, MSD analysis has a much larger standard deviation in estimated parameter values than the JDD in all cases studied. The JDD continues to improve upon the MSD as we have longer trajectories, more trajectories, and a shorter time lag, all of which lead to more data points. Relative performance of the two schemes was stable across parameter regimes explored.

With directed motion, it reflects that there is a time lag "sweet-spot" for best results (For more on this see S2 Table). MSD analysis performs particularly poorly when the drift term (V) is significantly larger than the diffusion term (D_V), as was the case in our analysis. In this case, MSD analysis cannot reliably extract the diffusion parameter. This leads us to another large advantage of JDD analysis; when the directed part of motion is much larger than the diffusive part, JDD analysis can reliably extract the diffusion constant, whereas MSD cannot. We expect this would also be the case in a combination model, where only a small fraction of particles are undergoing one type of motion, or in the case of two diffusion constants avoiding the averaging of the two.

Discussion

Particulate trajectory analysis is used in many different fields of study. While MSD analysis is easy to use, it often oversimplifies systems and with small amounts of data, can lead to inaccurate estimation of parameters. With increases in computing power and mathematical tools, a better and more versatile method should be used.

The aim of this paper has been to describe a general method for bringing trajectory analysis up to date. The JDD method overcomes the issue of small amounts of data compared to the MSD, and with large amounts of data, is just as accurate. It allows for selection between competing models, which is a major advantage when uncertain of the underlying behavior of the system. It can consider combination models, which MSD analysis cannot do. The general method is broadly applicable, it works for any dimension and any model, as long as the underlying JDD frequency distribution can be derived.

We have provided the framework for implementing this model with experimental data and have posted our code online with examples for all dimensions and models. In S6 Appendix we outline tips for the application to experimental data. We also have derived and compiled the JDD frequency distributions for three modes of motion in one, two, and three dimensions. This allows for the easy creation of combination models and the ability to examine three dimensional data anisotropically.

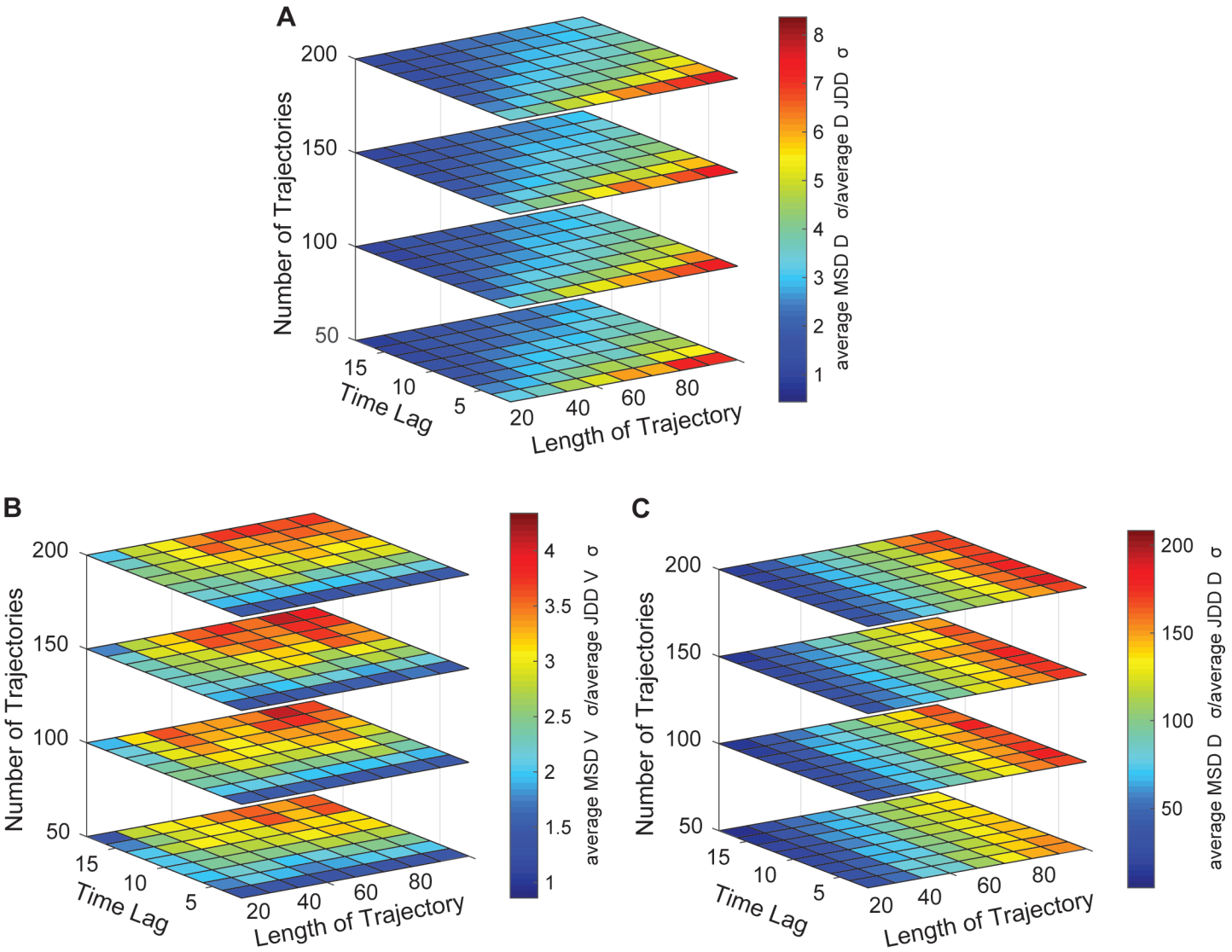

Fig 4. MSD vs. JDD For this figure, we analyzed the standard deviation of parameter values by using bootstrapping(which we use to define our error bounds) for both MSD and JDD. A: Average MSD σ/average JDD σ for pure diffusion for varying the length of the trajectory, the time lag, and the number of trajectories. Figure done for D=1 $\mu m^2/s$ and dt=1 s. B: Average MSD σ/average JDD σ for directed diffusion V parameter for varying the length of the trajectory, the time lag, and the number of trajectories. Figure for V=1 m/s , D=0.1 $\mu m^2/s$ and dt=1 s. C: Average MSD σ/average JDD σ for pure diffusion for varying the length of the trajectory, the time lag, and the number of trajectories. Figure for V=1 m/s, D=0.1 $\mu m^2/s$ and dt=1 s. The JDD is significantly better in this case because the system is so dominated by the directed motion that it greatly struggles to accurately determine the diffusion constant.

Supporting information

We have nine examples (for each model and each dimension) that walk through our analysis, from simulating the data to performing the parameter estimation and model selection. We believe this will be of great use to other researchers such that they do not have to write the code for themselves and just be able to use it to conduct analysis from an experimental JDD. Our code is open-source and can be found at `https://github.com/rmenssen/JDD_Code/`.

S1 Appendix. Closed form JDD Derivations Derivations are given for the closed form JDDs for pure diffusion, directed diffusion, and anomalous diffusion in one, two and three dimensions.

S2 Table. The Effect of time lag on parameter estimation Keeping the total number of data points the size, we examine how different time lags affect parameter estimation and model selection.

S3 Table. The Effect of number of bins on parameter estimation For each model, we keep everything constant, but then vary the number of bins in the JDD in order to understand the effect of bin size on parameter fitting and model selection.

S4 Table. Parameter Fitting Results for non-sliding vs sliding JDD construction-Diffusion In this table we show the results of parameter fitting for two ways to constructing data, showing that the sliding results are worse, but not very significantly given how much less data you need.

S5 Table. Parameter Fitting Results for non-sliding vs sliding JDD construction-Directed Motion In this table we show the results of parameter fitting for two ways to constructing data, showing that the sliding results are worse, but not very significantly given how much less data you need.

S6 Appendix. Tips for Practical Application In this section, we discuss various considerations that need to be made to apply our method to experimental data. This section also goes in depth into some of the numerical considerations we had to make with simulated data that are also helpful for experimental data.

Acknowledgements

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1324585.

References

1. Tollis S. A Jump Distance-based Bayesian analysis method to unveil fine single molecule transport features. arXiv preprint arXiv:150601112. 2015;.
2. Chandrasekhar S. Stochastic problems in physics and astronomy. Reviews of modern physics. 1943;15(1):1–89.
3. Blackburn N, Fenchel T, Mitchell J. Microscale nutrient patches in planktonic habitats shown by chemotactic bacteria. Science. 1998;282(5397):2254–2256.

4. Taktikos J, Stark H, Zaburdaev V. How the motility pattern of bacteria affects their dispersal and chemotaxis. PloS one. 2013;8(12):e81936.

5. Garcia HG, Tikhonov M, Lin A, Gregor T. Quantitative imaging of transcription in living Drosophila embryos links polymerase activity to patterning. Current biology. 2013;23(21):2140–2145.

6. Qian H, Sheetz MP, Elson EL. Single particle tracking. Analysis of diffusion and flow in two-dimensional systems. Biophysical journal. 1991;60(4):910–921.

7. Saxton MJ, Jacobson K. Single-particle tracking: applications to membrane dynamics. Annual review of biophysics and biomolecular structure. 1997;26(1):373–399.

8. Marquez-Lago T, Leier A, Burrage K. Anomalous diffusion and multifractional Brownian motion: simulating molecular crowding and physical obstacles in systems biology. IET systems biology. 2012;6(4):134–142.

9. Michalet X. Mean square displacement analysis of single-particle trajectories with localization error: Brownian motion in an isotropic medium. Physical Review E. 2010;82(4):041914.

10. Türkcan S, Masson JB. Bayesian decision tree for the classification of the mode of motion in single-molecule trajectories. PloS one. 2013;8(12):e82799.

11. Monnier N, Barry Z, Park HY, Su KC, Katz Z, English BP, et al. Inferring transient particle transport dynamics in live cells. Nature methods. 2015;12(9):838–840.

12. Kepten E, Weron A, Sikora G, Burnecki K, Garini Y. Guidelines for the fitting of anomalous diffusion mean square displacement graphs from single particle tracking experiments. PLoS One. 2015;10(2):e0117722.

13. Burnecki K, Kepten E, Garini Y, Sikora G, Weron A. Estimating the anomalous diffusion exponent for single particle tracking data with measurement errors-An alternative approach. Scientific reports. 2015;5.

14. Meroz Y, Sokolov IM. A toolbox for determining subdiffusive mechanisms. Physics Reports. 2015;573:1–29.

15. Monnier N, Guo SM, Mori M, He J, Lénárt P, Bathe M. Bayesian approach to MSD-based analysis of particle motion in live cells. Biophysical journal. 2012;103(3):616–626.

16. Wu J, Berland KM. Propagators and time-dependent diffusion coefficients for anomalous diffusion. Biophysical journal. 2008;95(4):2049–2052.

17. Kues T, Dickmanns A, Lührmann R, Peters R, Kubitscheck U. High intranuclear mobility and dynamic clustering of the splicing factor U1 snRNP observed by single particle tracking. Proceedings of the National Academy of Sciences. 2001;98(21):12021–12026.

18. Anderson CM, Georgiou GN, Morrison I, Stevenson G, Cherry RJ. Tracking of cell surface receptors by fluorescence digital imaging microscopy using a charge-coupled device camera. Low-density lipoprotein and influenza virus receptor mobility at 4 degrees C. Journal of cell science. 1992;101(2):415–425.

19. Grünwald D, Martin RM, Buschmann V, Bazett-Jones DP, Leonhardt H, Kubitscheck U, et al. Probing intranuclear environments at the single-molecule level. Biophysical journal. 2008;94(7):2847–2858.

20. Weimann L, Ganzinger KA, McColl J, Irvine KL, Davis SJ, Gay NJ, et al. A quantitative comparison of single-dye tracking analysis tools using Monte Carlo simulations. PloS one. 2013;8(5):e64287.

21. Siebrasse JP, Veith R, Dobay A, Leonhardt H, Daneholt B, Kubitscheck U. Discontinuous movement of mRNP particles in nucleoplasmic regions devoid of chromatin. Proceedings of the National Academy of Sciences. 2008;105(51):20291–20296.

22. Metzler R, Klafter J. The random walk's guide to anomalous diffusion: a fractional dynamics approach. Physics reports. 2000;339(1):1–77.

23. Montroll EW, Scher H. Random walks on lattices. IV. Continuous-time walks and influence of absorbing boundaries. Journal of Statistical Physics. 1973;9(2):101–135.

24. Crank J. The mathematics of diffusion. Oxford university press; 1979.

25. Marquardt DW. An algorithm for least-squares estimation of nonlinear parameters. Journal of the society for Industrial and Applied Mathematics. 1963;11(2):431–441.

26. Moré JJ. The Levenberg-Marquardt algorithm: implementation and theory. In: Numerical analysis. Springer; 1978. p. 105–116.

27. Efron B, Tibshirani RJ. An introduction to the bootstrap. CRC press; 1994.