Bayesian Approach to Parameter Estimation in Individual Protein Molecule Dynamics Model

M.Malyutov
and T. Korobeinikova
Department of Mathematics
Northeastern University
Boston, MA 02115,
USA

R.S. Protassov
Statistics Department
Harvard University,
1 Oxford St.,
Cambridge, MA 02138,
USA

D. Golan and R. Mirchev
Dept. of Biological Chemistry
and Molecular Pharmacology,
Harvard Medical School,
Boston, MA 02115,
USA

January 20, 2002

Abstract: - A model for the dynamics of membrane proteins in intact red blood cells was proposed in [3]. The data are obtained with a CCD camera. A nm-scale gold bead is attached to a protein molecule. Differential interference contrast (DIC) video microscopy is used to image the bead, and the images are recorded by a high-speed CCD camera. Because the beads are smaller than the resolution limit of the microscope, the bead images appear as blurred spots. The centroid position of the bead in each image is determined by using a tracking algorithm.

In [3] the motion of an individual protein is modeled as a sum of three mutually independent components: a Wiener process, a compound Poisson process, and white Gaussian noise. This approach allows modeling of the interactions of the protein with various types of molecules. Model parameters were estimated by the method of moments applied to each of 10 equal parts of the rajectory. Thus we obtained a histogram for the MOM parameter estimates. In this paper, we propose Bayesian parameter estimation based on Gibbs sampler. The uncertainty in parameter values is quantitfied via credible intervals (see [1]).

Keywords: - Protein dynamics model, Bayesian sampling, Markov Chain Monte Carlo.

I Introduction

This paper is based on our studies of the dynamics of individual protein molecules in the membranes of intact red blood cells [7]. A gold bead (5 to 40 nm) is attached to the molecule, and plane images (typically 128 by 80 pixels) of the molecule's trajectory are recorded using DIC video microscopy at a rate of 30 to 10,000 frames per second.

Heuristic threshold- or moment-based algorithms often used by experimenters, besides being inaccurate in strong noise, are heavily biased when bead spots overlap and/or when a spot is too large or bead is too close to the image edge to fit a spot into the image frame due to the limited view area of the microscope.

We proposed in [6] a procedure based on the EM algorithm which estimates the not centrally symmetrical bead position more accurately adapting to possible rotations. The main idea of the approach is to estimate parameters relevant to the bead spot generation using a series of images, and then to solve the inverse problem by establishing the most probable center coordinate for each image along with other characteristics (variance etc).

Traditionally, the motion of individual protein molecules is described as a Wiener process (Brownian motion) or a Wiener process with additive Gaussian white noise (see [2]). Typically, the Mean Square Distance statistic (MSD) is used to estimate the motion parameter. An alternative estimation technique has been proposed in [4], in which a maximum likelihood (ML) estimate is evaluated via spectral decomposition of the process covariance matrix.

Experimental data for the motion of a variety of proteins show considerable deviation from the Brownian motion model, causing instability in the estimated parameters of motion. Thus, even for small time intervals, some protein molecules do not appear to undergo free diffusion (see, for example, [8]). Free diffusion of red cell membrane proteins could be restricted due to interactions between the protein and a) molecules of different sizes, e.g. lipids and other membrane proteins; and b) the spectrin-based membrane skeleton. A detailed physical model of this phenomenon remains to be developed.

Model developed in [3] describes the effect of protein interactions with other membrane proteins (i.e. with large particles) as a compound Poisson process with normally distributed jumps. The assumption is that protein's interactions with large particles are rare compared to its collisions with small particles (i.e. lipids).

In this paper, we propose Bayesian parameter estimation for the model. This method allows to quantify uncertainty in parameter estimates via credible intervals (see e.g. [1]).

II Model Specification

Suppose we observe a random sequence $\mathbf{Y} = (Y(t_i), t_i = iT/N, i = 0, ..., N)$ with independent increments, corrupted by white Gaussian noise:

$$Y(t_i) = aW(t_i) + b\sum_{j=1}^{\pi(t_i)} \eta_j + c\varepsilon(t_i), \qquad (1)$$

where a, b, c are positive constants; $W(\cdot)$ is a standard Wiener process; η_j $(j = 1, \ldots, \pi(t_i))$ are *i.i.d.* standard normal variables; $\pi(\cdot)$ is a Poisson process with rate μ; and $\varepsilon(\cdot)$ is standard white Gaussian noise. All four sources of randomness, $W(\cdot)$, $\pi(\cdot)$, η_j $(j = 1, \ldots, \pi(t_i))$, and $\varepsilon(\cdot)$ are mutually independent.

In this formula $aW(\cdot)$ represents displacements due to interactions between the labeled protein and membrane lipids, $b\sum_{j=1}^{\pi(t_i)} \eta_j$ describes displacements due to interactions between the protein and other membrane proteins, and $c\varepsilon(\cdot)$ represents measurement noise. We let θ denote parameters to be estimated:

$$\theta = (a, b, c, \mu). \qquad (2)$$

III Bayesian Formulation

In Bayesian approach, the parameters $\theta = (a, b, c, \mu)$ themselves are treated as random variables and the goal of Bayesian estimation is to describe the posterior distribution of the parameters (that is, the distribution of the parameters given the data $Y(t_i)$). This can only be achieved if a prior distribution is specified for θ. In picking a prior we are guided by the following two considerations. First of all, we want to be able to tune the hyperparameters (that is, the parameters of the prior distribution) to make the prior practically as flat as possible over the range of plausible values for θ. Secondly, the prior should be of the form that will be computationally convenient. We will coment on this second point at the end of this section.

We consider the incremets $X(t_i)$ of the process (1) as the observed data. These increments are representable as

$$\begin{aligned} X(t_i) &= a\Delta W(t_i) + b\sum_{j=\pi(t_{i-1})+1}^{\pi(t_i)} \eta_j + c\Delta\epsilon(t_i) \qquad (3) \\ &\sim a\Delta W(t_i) + b\sum_{j=1}^{\Delta\pi(t_i)} \eta_j + c\Delta\epsilon(t_i), \end{aligned}$$

$t_i = iT/N, \quad i = 1, \ldots, N$, where

$$\begin{aligned} \Delta W(t_i) &= W(t_i) - W(t_{i-1}), \\ \Delta\pi(t_i) &= \pi(t_i) - \pi(t_{i-1}), \\ \Delta\epsilon(t_i) &= \epsilon(t_i) - \epsilon(t_{i-1}). \end{aligned}$$

We postulate the following prior on θ:

$$\begin{aligned} a^{-2} &\sim \text{Gamma}(\alpha_a, \beta_a), \qquad (4) \\ b^{-2} &\sim \text{Gamma}(\alpha_b, \beta_b), \\ c &\sim \text{N}(m_c, \sigma_c^2),\ c > 0 \text{ (truncated normal)}, \\ \mu &\sim \text{Gamma}(\alpha_\mu, \beta_\mu). \end{aligned}$$

We assume that that the priors are independent.

Here the parameters of the priors α_a, β_a, α_b, β_b, m_c, σ_c, α_μ, β_μ (the so-called hyper parameters) are constants known

in advance. We choose these constants so that the priors look reasonably flat over the range that we believe almost certainly contains the true parameter values.

The posterior is too complicated to be amenable to direct examination, we thus resort to studying the posterior by producing a sample from it. This sample, in turn, cannot be obtained directly. Instead we will construct a Markov chain that will converge to the posterior of interest and will simulate the parameters with the help of this chain. More specifically, we propose a Gibbs sampler combined with a "data augmentation" technique [10]. A good description of the data augmentation technique could be also found in [9]. Now, the computational convenience we referred to in the begining of this section should be understood as prior specification that simplifies steps of the Gibbs sampler.

IV Gibbs Sampler.

To simplify the notations, we will use set the time interval between observations to be equal to 1 ($T = N$). Define

$$\begin{aligned} X &= (X_1, \ldots, X_N) = (X(t_1), \ldots, X(t_N)), \\ \Delta W &= (\Delta W_1, \ldots, \Delta W_N) = (\Delta W(t_1), \ldots, \Delta W(t_N)), \\ \Delta\pi &= (\Delta\pi_1, \ldots, \Delta\pi_N) = (\Delta\pi(t_1), \ldots, \Delta\pi(t_N)), \\ \Delta\epsilon &= (\Delta\epsilon_1, \ldots, \Delta\epsilon_N) = (\Delta\epsilon(t_1), \ldots, \Delta\epsilon(t_N)). \end{aligned}$$

Using the language of data augmentation technique [10], we call X the observed data and $\Delta\pi, \Delta\epsilon$ the missing data or the unobserved variables. Together they make up what is referred to as "augmented data".

We start the Gibbs sampling procedure by setting some starting values for the unobserved variables. We then repeatedly visit each unobserved variable and parameter in turn, each time randomly selecting a new value for the variable from its conditional distribution given the current values of the other variables and parameters. Schematically, our Gibbs sampler can be represented as follows:

0. Pick a starting value for the parameters θ^0 and for the vector with the number of jumps $\Delta\pi^0$ (to simplify the notations, we will sometimes use $\theta = (a,\ b,\ c,\ \mu)$).
1. Simulate $\Delta\epsilon|\theta^0, X, \Delta\pi^0$ to obtain $\Delta\epsilon^1$.
2. Simulate $\Delta\pi|\Delta\epsilon^1, X, \theta^0$ to obtain $\Delta\pi^1$.
3. Simulate $\theta|\Delta\epsilon^1, \Delta\pi^1, X$ to obtain θ^1.

3.1 Simulate $\mu|\Delta\epsilon^1, \Delta\pi^1, X, a^0, b^0, c^0$ to obtain μ^1.

3.2 Simulate $a|\Delta\epsilon^1, \Delta\pi^1, X, b^0, c^0, \mu^1$ to obtain a^1.

3.3 Simulate $b|\Delta\epsilon^1, \Delta\pi^1, X, a^1, c^0, \mu^1$ to obtain b^1.

3.4 Simulate $c|\Delta\epsilon^1, \Delta\pi^1, X, a^1, b^1, \mu^1$ to obtain c^1.

4. Set $\theta^1,\ \Delta\pi^1$ to be the new starting values and repeat the steps 1–3.

This is a Makov chain with the posterior of interest as its stationary distribution. Implementation of this Gibbs sampler depends on our ability to simulate from the conditional distributions listed in steps 1–3. We derive these conditional distributions and provide algorithm to generate random variates from them below.

A Details of the Gibbs Sampler

We can pick any a priory reasonable values for $a,\ b,\ c$, and μ as the initial values. We can then initialize the unobserved variates $\Delta\pi$ by taking *i.i.d.* draws from $\Delta\pi_i^{init} \sim \text{Poiss}(\mu_{init})$. Our starting value for the Gibbs sampler thus will be $(\theta^0, \Delta\pi^0) = (a_{init}, b_{init}, c_{init}, \mu_{init}, \Delta\pi^{init})$.

In Step 1 we draw $\Delta\epsilon$ using the classical Backward-Forward calculations. The Forward Calculations are represented schematically as

$$\begin{aligned} \text{Find } P(\Delta\epsilon_1|X_1, \Delta\pi, \theta) &\to P(\Delta\epsilon_2|X_1, X_2, \Delta\pi, \theta) \\ &\to \cdots \to P(\Delta\epsilon_N|X_1, \ldots, X_N, \Delta\pi, \theta). \end{aligned}$$

Given that we derived $P(\Delta\epsilon_N|X_1, \ldots, X_N, \Delta\pi, \theta)$, we obtain a random draw of $\Delta\epsilon_N$ from this distribution and then go backwards by drawing $\Delta\epsilon_j$ from $P(\Delta\epsilon_j|X_1, \ldots, X_N, \Delta\pi, \Delta\epsilon_N, \ldots, \Delta\epsilon_{j+1}, \theta)$:

$$\begin{aligned} \text{Draw } \Delta\epsilon_N|X, \Delta\pi, \theta &\to \\ \text{Find } P(\Delta\epsilon_{N-1}|X, \Delta\pi, \Delta\epsilon_N, \theta) &\to \\ \text{Draw } \Delta\epsilon_{N-1}|X, \Delta\pi, \Delta\epsilon_N, \theta &\to \cdots \to \\ \text{Find } P(\Delta\epsilon_1|X, \Delta\pi, \Delta\epsilon_N, \ldots, \Delta\epsilon_2, \theta) &\to \\ \text{Draw } \Delta\epsilon_1|X, \Delta\pi, \Delta\epsilon_N, \ldots, \Delta\epsilon_2, \theta. & \end{aligned}$$

With the final draw of $\Delta\epsilon_1$ in this procedure we will have a draw from the joint distribution $P(\Delta\epsilon|X, \Delta\pi, \theta)$.[1]

What makes these manipulations possible is the fact that all the distributions P that appear in Forward-Backward calculations are normal. Indeed, in the forward calculations:

$$\begin{aligned} P(\Delta\epsilon_1|X_1, \Delta\pi) &= \frac{P(\Delta\epsilon_1, X_1|\Delta\pi_1)}{P(X_1|\Delta\pi_1)} \\ P(X_1|\Delta\pi_1) &= \mathrm{N}(X_1|0, a^2 + \Delta\pi_1 b^2 + 2c^2), \end{aligned}$$

where $\mathrm{N}(X|m, \sigma^2)$ stands for the density of normal distribution with mean m and variance σ^2 evaluated at point X. Then

$$\begin{aligned} P(\Delta\epsilon_1, X_1|\Delta\pi_1) &= P(X_1|\Delta\epsilon_1, \Delta\pi_1)P(\Delta\epsilon_1|\Delta\pi_1) \\ &= \mathrm{N}(X_1|c\Delta\epsilon_1, a^2 + \Delta\pi_1 b^2) \\ &\quad \times \mathrm{N}(\Delta\epsilon_1|0, 2). \end{aligned}$$

Thus,

$$\begin{aligned} P(\Delta\epsilon_1|X_1, \Delta\pi_1) &= \frac{\mathrm{N}(X_1|c\Delta\epsilon_1, a^2 + \Delta\pi_1 b^2)\mathrm{N}(\Delta\epsilon_1|0, 2)}{\mathrm{N}(X_1|0, a^2 + \Delta\pi_1 b^2 + 2c^2)} \\ &= \mathrm{N}(\Delta\epsilon_1|m_{1,f}, \sigma^2_{1,f}), \end{aligned}$$

where

$$\begin{aligned} \sigma^2_{1,f} &= \left[\frac{a^2 + \Delta\pi_1 b^2 + 2c^2}{2(a^2 + \Delta\pi_1 b^2)}\right]^{-1}, \\ m_{1,f} &= \frac{cX_1}{a^2 + \Delta\pi_1 b^2}\sigma^2_{1,f} = \frac{2cX_1}{a^2 + \Delta\pi_1 b^2 + 2c^2}. \end{aligned}$$

We proceed by induction. Suppose we have derived:

$$P(\Delta\epsilon_j|X_1, \ldots, X_j, \Delta\pi) = \mathrm{N}(\Delta\epsilon_j|m_{j,f}, \sigma^2_{j,f})$$

with $m_{j,f}$ and $\sigma^2_{j,f}$ known. We want to derive $P(\Delta\epsilon_{j+1}|X_1, \ldots, X_{j+1}, \Delta\pi)$. Since

$$\begin{aligned} P(\Delta\epsilon_j, \Delta\epsilon_{j+1}, X_{j+1}|X_1, \ldots, X_j, \Delta\pi) = \\ P(X_{j+1}|\Delta\epsilon_{j+1}, \Delta\pi_{j+1})P(\Delta\epsilon_{j+1}|\Delta\epsilon_j) \\ \times P(\Delta\epsilon_j|X_1, \ldots, X_j, \Delta\pi), \end{aligned}$$

where the three distributions are:

$$\begin{aligned} P(X_{j+1}|\Delta\epsilon_{j+1}, \Delta\pi_{j+1}) &= \mathrm{N}(X_{j+1}|c\Delta\epsilon_{j+1}, a^2 + \Delta\pi_{j+1}b^2), \\ P(\Delta\epsilon_{j+1}|\Delta\epsilon_j) &= \mathrm{N}(\Delta\epsilon_{j+1}|-\frac{1}{2}\Delta\epsilon_j, \frac{3}{2}), \\ P(\Delta\epsilon_j|X_1, \ldots, X_j, \Delta\pi) &= \mathrm{N}(\Delta\epsilon_j|m_{j,f}, \sigma^2_{j,f}), \end{aligned}$$

we can show that

$$P(\Delta\epsilon_{j+1}|X_1, \ldots, X_{j+1}, \Delta\pi) = \mathrm{N}(\Delta\epsilon_{j+1}|m_{j+1,f}, \sigma^2_{j+1,f})$$

with

$$\begin{aligned} \sigma^2_{j+1,f} &= \left[\frac{c^2}{a^2 + \Delta\pi_{j+1}b^2} + \frac{4}{\sigma^2_{j,f} + 6}\right]^{-1} \\ m_{j+1,f} &= \left[\frac{cX_{j+1}}{a^2 + \Delta\pi_{j+1}b^2} - \frac{2m_{j,f}}{\sigma^2_{j,f} + 6}\right]\sigma^2_{j+1,f}. \end{aligned}$$

By analogy, having drawn $\Delta\epsilon_N$ from $P(\Delta\epsilon_N|X, \Delta\pi)$, we move backwards: given $\Delta\epsilon_N, \ldots, \Delta\epsilon_j$ we draw $\Delta\epsilon_{j-1}$ from

$$\begin{aligned} P(\Delta\epsilon_{j-1}|X, \Delta\pi, \Delta\epsilon_N, \ldots, \Delta\epsilon_j) = \\ \mathrm{N}(\Delta\epsilon_{j-1}|m_{j-1,b}, \sigma^2_{j-1,b}), \end{aligned}$$

with

$$\begin{aligned} \sigma^2_{j-1,b} &= \left[\frac{1}{6} + \frac{1}{\sigma^2_{j-1,f}}\right]^{-1}, \\ m_{j-1,b} &= \left[\frac{m_{j-1,f}}{\sigma^2_{j-1,f}} - \frac{\Delta\epsilon_j}{3}\right]\sigma^2_{j-1,b}. \end{aligned}$$

In Step 2, we notice that $\Delta\pi_i|X, \Delta\epsilon$ are independent since the probability mass function is

$$P(\Delta\pi|X, \Delta\epsilon) = \prod_{i=1}^{N} P(\Delta\pi_i|X_i, \Delta\epsilon_i),$$

where

$$\begin{aligned} P(\Delta\pi_i = j|X_i, \Delta\epsilon_i) = \\ \frac{\frac{\mu^j}{j!}e^{-\mu}\mathrm{N}(X_i|c\Delta\epsilon_j, a^2 + jb^2)}{\sum_{l=0}^{+\infty}\frac{\mu^l}{l!}e^{-\mu}\mathrm{N}(X_i|c\Delta\epsilon_j, a^2 + lb^2)}. \end{aligned}$$

In Step 3.1 it is straightforward to verify that

$$\mu|\Delta\pi \sim \mathrm{Gamma}(\alpha_\mu + \sum_{i=1}^{N}\Delta\pi_i, \beta_\mu + N).$$

To draw parameters a and b in Steps 3.2-3.3 we will use Metropolis-Hastings algorithm (see, e.g., [1]). In general, to produce a draw of a variate ξ from target distribution $P(\xi)$, whose

[1] In all the conditional formulas we also assume conditioning on the current parameter estimates, though sometimes we do not write this explicitly.

density we can compute up to a normalizing constant, the Metropolis-Hastings algorithm creates a sequence of random points $\xi^1, \xi^2, \ldots$ whose distribution converge to the target distribution. Each sequence can be considered as a random walk with stationary distribution $P(\xi)$. Below we list the steps of the algorithm:
1. Draw a starting point ξ^0, for which $P(\xi^0) > 0$, from a starting distribution $P_0(\xi)$.
2. For $t = 1, 2, \ldots$:
(a) Sample a candidate point ξ^* from a jumping distribution at time t, $J_t(\xi^*|\xi^{t-1})$ (we use the jumping distribution to approximate the target distribution).
(b) Calculate the ratio of importance ratios:

$$r = \frac{P(\xi^*|X)J_t(\xi^*|\xi^{t-1})}{P(\xi^{t-1})J_t(\xi^{t-1}|\xi^*)}$$

(c) Set $\xi^t = \xi^*$ with probability $\min(r, 1)$, and $\xi^t = \xi^{t-1}$ with probability $1 - \min(r, 1)$

In Step 3.2, for parameter a the target bdistribution is

$$P(a^{-2}|\Delta\epsilon, \Delta\pi, X) \propto P(a^{-2}) \times \prod_{i=1}^{N}(a^2 + \Delta\pi_i b^2)^{-1/2} \exp\left(-\frac{(X_i - c\Delta\epsilon_i)^2}{2(a^2 + \Delta\pi_i b^2)}\right)$$

where $P(a^{-2})$ is the prior distribution

$$P(a^{-2}) \propto \left(a^{-2}\right)^{\alpha_a - 1} \exp\left(-a^{-2}\beta_a\right).$$

The jumping distribution:

$$P_{jump}(a^{-2}|\Delta\epsilon, \Delta\pi, X) \propto \prod_{i=1}^{N}(a^2)^{-1/2} \exp\left(-\frac{(X_i - c\Delta\epsilon_i)^2}{2a^2}\right) P(a^{-2}) \propto \text{Gamma}(\alpha_a + \frac{N}{2}, \beta_a + \frac{1}{2}\sum_{i=1}^{N}(X_i - c\Delta\epsilon_i)^2)$$

In Step 3.3, for b the target is:

$$P(b^{-2}|\Delta\epsilon, \Delta\pi, X) \propto P(b^{-2}) \times \prod_{i=1}^{N}(a^2 + \Delta\pi_i b^2)^{-1/2} \exp\left(-\frac{(X_i - c\Delta\epsilon_i)^2}{2(a^2 + \Delta\pi_i b^2)}\right)$$

where $P(b^{-2})$ is the prior distribution:

$$P(b^{-2}) \propto \left(b^{-2}\right)^{\alpha_b - 1} \exp\left(-b^{-2}\beta_b\right).$$

and jumping distribution:

$$P_{jump}(b^{-2}|\Delta\epsilon, \Delta\pi, X) \propto \prod_{i=1,\Delta\pi>0}^{N}(b^2)^{-1/2} \exp\left(-\frac{(X_i - c\Delta\epsilon_i)^2}{2\Delta\pi_i b^2}\right) P(b^{-2}) \propto \text{Gamma}\left(\alpha_b + \sum_{i=1,\Delta\pi>0}^{N}\frac{1}{2}, \beta_a + \frac{1}{2}\sum_{i=1,\Delta\pi>0}^{N}\frac{(X_i - c\Delta\epsilon_i)^2}{\Delta\pi_i}\right)$$

In Step 3.4 it is easily verified that c is truncated normal

$$\begin{aligned} c|X, c\Delta\epsilon, \Delta\pi, b &\sim \text{N}(m_{c_{new}}, \sigma^2_{c_{new}}), \\ \text{where} \quad \sigma^2_{c_{new}} &= \left(\frac{1}{\sigma_c^2} + \frac{\Delta\epsilon_i^2}{a^2 + \Delta\pi_i b^2}\right)^{-1} \\ \text{and} \quad m_{c_{new}} &= \left(\frac{m_c}{\sigma_c^2} + \sum_{i=1}^{N}\frac{\Delta\epsilon_i X_i}{a^2 + \Delta\pi_i b^2}\right)\sigma^2_{c_{new}}, \end{aligned}$$

Thus the description of the Gibbs samapler steps is complete.

V Parameter Estimates for Experimental Data

We studied the dynamics of CD58 based on a trajectory consisting of 10,000 frames obtained at the rate of 1,000 frames per second. The x and y coordinates of the protein trajectories were extracted from images using a tracking algorithm. Below we provide the results of parameter estimation via the method of moments (MOM) developed in [3] and the credible interval obtained with the Bayesian method described in this article.

For the Bayesian method the choice of the prior parameter distribution was the following:

$$\begin{aligned} a^{-2} &\sim \text{Gamma}(0.1, 10), \\ b^{-2} &\sim \text{Gamma}(0.04, 20), \\ c &\sim \text{N}(10, 5), c > 0(\text{truncated normal}), \\ \mu &\sim \text{Gamma}(1, 1). \end{aligned} \tag{5}$$

Notice that with this choice of prior distributions the probability density functions for a and b are the following:

$$P_a(x) = \frac{1}{2}x^{-\frac{3}{2}}P_{\mathrm{Gamma}_{\mathrm{a}-2}}(x^{-\frac{1}{2}})$$
$$P_b(x) = \frac{1}{2}x^{-\frac{3}{2}}P_{\mathrm{Gamma}_{\mathrm{b}-2}}(x^{-\frac{1}{2}}),$$

where $P_{\mathrm{Gamma}_{\mathrm{a}-2}}(.)$, $P_{\mathrm{Gamma}_{\mathrm{b}-2}}(.)$ are the densities of the Gamma distributions (in our case Gamma$(0.1, 10)$ and Gamma$(0.02, 20)$ respectively).

Table 1 *Parameter Estimates for Two-dimensional Protein CD58 Trajectory* X-coordinate:

	a	b	c	μ
MOM	7.7	22.7	9.67	0.018
Bayesian 95% credible interval	[6.3; 8.3]	[22.3; 51.1]	[9.6; 10.8]	[0.001; 0.022]

Y-coordinate:

	a	b	c	μ
MOM	7.7	37.0	10.3	0.014
Bayesian 95% credible interval	[6.5; 8.3]	[25.0; 36.5]	[10.2; 11.0]	[0.013; 0.044]

References

[1] Gelman, A., Carlin, J., Stern, H., and Rubin, D. (2000): *Bayesian Data Analysis.* Chapman & Hall/CRC.

[2] Kusumi, A., Sako, Y. and Yamamoto, M. (1993): Confined Lateral Diffusion of Membrane Receptors as Studied by Single Particle Tracking. *Biophysical Journal*, *Vol. 65*, pp. 2021–2040.

[3] Malyutov, M. *et al.* (2001): Modeling and tracking of the Individual Protein Molecule Dynamics. In *G. Govaert ed.: Proceedings of ASMDA-10 (Applied Stochastic Modeling and Data Analysis) , Compiegne, France, June 12-15, 2001, Vol. 2*, pp. 627-632.

[4] Malyutov, M. and Bayborodin, O. (1999): Estimation of Brownian Volatility in Noise via Mercer Expansion. In: V. Vishnevsky and E. Levner ed.:*Proceedings of the International Conference on Distributed Computer Communication Networks*. November 9–13, Tel-Aviv, Israel, pp. 89–93.

[5] Malyutov, M., Bayborodin, O., Mirchev, R. and Golan, D. (2000): Fitting Diffusion and Trend via Mercer Expansion. In: D. Moeller ed.:*Proceedings, 12th European Simulation Symposium, ESS 2000 (Simulation in Industry)*. September 28–30, Hamburg, Germany, Soc. for Computer Simulation, Delft, Netherlands, pp. 646–650.

[6] Malyutov, M., Nikiforov, A., Mirchev, R. and Golan, D. (2000): Estimation from Noisy Images with the EM-algorithm. In: Proceedings of the International Conference on Statistics, Combinatorics and related fields, December 2000, Mumbai, India, to appear.

[7] Mirchev, R., Golan, D. (2001): Single-particle Tracking and Laser Optical Tweezers Studies of the Dynamics of Individual Protein Molecules in Membranes of Intact Human and Mouse Red Cells. *Blood Cells Molecules Diseases*, *Vol. 27*, 143–147.

[8] Saxton, M., and Jacobson, K. (1999): Single Particle Tracking: Applications to Membrane Dynamics. *Annual Review Biophys. Biomol. Struct.*, *Vol. 26*, pp. 373–399.

[9] Schafer, J. L. (1997): *Analysis of Incomplete Multivariate Data.* Chapman & Hall, New York.

[10] Tanner, M. A., and Wong, W. H. (1987): The Calculation of Posterior Distributions by Data Augmentation (with discussion). *Journal of the American Statistical Association*, *Vol. 82*, pp. 528–550.