

Mixtures of Self-Modelling Regressions

Rhonda D Szczesniak^{1*}, Kert Viele^{2,3} and Robin L Cooper⁴

¹Division of Biostatistics and Epidemiology, Cincinnati Children's Hospital Medical Center, USA

²Berry Consultants, LLC, USA

³Department of Statistics, University of Kentucky, USA

⁴Department of Biology, University of Kentucky, USA

Abstract

A shape invariant model for functions f_1, \dots, f_n specifies that each individual function f_i can be related to a common shape function g through the relation $f_i(x) = a_i g(c_i x + d_i) + b_i$. We consider a flexible mixture model that allows multiple shape functions g_1, \dots, g_K , where each f_i is a shape invariant transformation of one of those g_k . We derive an MCMC algorithm for fitting the model using Bayesian Adaptive Regression Splines (BARS), propose a strategy to improve its mixing properties and utilize existing model selection criteria in semiparametric mixtures to select the number of distinct shape functions. We discuss some of the computational difficulties that arise. The method is illustrated using synaptic transmission data, where the groups of functions may indicate different active zones in a synapse.

Introduction

Self-Modeling regressions form a class of models [1,2] for functional data observed for many individuals. We observe data, where the subject specific functions $f_i(x)$ (sometimes referred to as traces) have the form where g is a base function and θ_i is a subject specific parameter which specifies a transformation connecting g and the functions f_i . A variety of choices is available for this transformation class which unites the subject specific functions f_i to g . One of the first examples in the literature, which we will address here, is the shape invariant model, referred to as SIM (1), where $\theta_i = (a_i, b_i, c_i, d_i)$ are called self-modeling coefficients and

$$f_i(x) = a_i g(c_i x + d_i) + b_i$$

The self-modeling coefficients result in f_i being an affine transformation of g in the x and y axes. Other types of transformations have been shown which use a Bayesian warping function method [3]. Altman and Villarreal [4] have developed the SIM where interest centers on variation in the self-modeling coefficients, using nonlinear mixed effects [5] and p-splines to model the θ_i parameters. More recent work involved inferential comparisons on θ_i in the SIM setting [6]. In this work, our primary objectives are the following:

1. To consider mixtures of shape invariant models (MSIMs) where there is more than one underlying shape function g . Unknown to the investigator, suppose the subject specific functions may be grouped so that one group is defined by shape invariant transformations of an underlying g_1 , another group is defined by shape invariant transformation of an underlying g_2 , and so on. A motivating example follows below.

2. To utilize Bayesian Adaptive Regression Splines (BARS) [7-9] to estimate the underlying shape functions g_k for $k=1, \dots, K$. BARS has been shown to provide parsimonious fits (e.g. fewer knots) of complicated functions. Several computational challenges arise in these endeavors which are described in Implementation.

In our motivating example, data are obtained from a recording of synaptic transmission at a neuromuscular junction (NMJ) from a crayfish (Figure 1a). The motor nerve is repeatedly stimulated and the synaptic response is monitored as described previously in work by two of the authors [10]. With low frequency stimulation of this low output NMJ, occasionally an evoked response is observed. In the sample shown, 61 events occurred in 1000 stimulations. After some data

cleaning [11] our data of interest is the 61 functions describing voltage over a dense series of time points.

These functions vary in terms of their heights, widths, and peak locations, but in fact do all have similar shapes as shown by the aligned curves which result from fitting a SIM model to all the functions (Figure 1b). The details of this fit, which simply assumed known knots for a

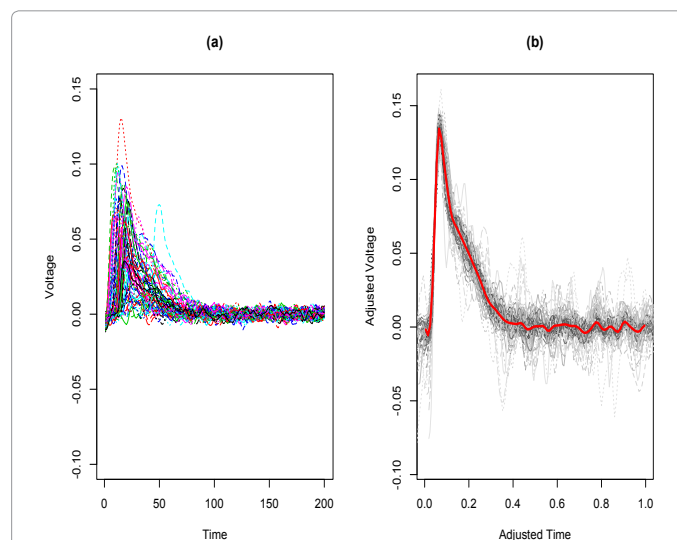


Figure 1: A set of 61 evoked excitatory post-synaptic potentials (EPSPs) Data in the left pane (a) describe the evoked voltage responses observed at the crayfish neuromuscular junction from 1000 stimulation trials. While the functions vary in height, width, and peak locations, all the functions have a similar underlying shape. The right pane (b) shows aligned firings with the estimated underlying shape g in red. These are the same data from (a) aligned to each other.

***Corresponding author:** Rhonda D Szczesniak, Division of Biostatistics & Epidemiology, Cincinnati Children's Hospital Medical Center, USA, Tel: 513-803-0563; E-mail: rhonda.szczesniak@cchmc.org

Received July 02, 2014; **Accepted** August 08, 2014; **Published** August 15, 2014

Citation: Szczesniak RD, Viele K, Cooper RL (2014) Mixtures of Self-Modelling Regressions. J Biom Biostat S12: 003. doi:10.4172/2155-6180.S12-003

Copyright: © 2014 Szczesniak RD, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

spline fit to g , are described in previous work [10]. This fit accounts for over 96% of the variation in the dataset.

Biologically, the variation between the functions is assumed to occur because of aspects related to the physiology and structure of the synapses at the NMJ. The electrical signal in the neuron is translated into a vesicle fusing with the presynaptic membrane at active zones to release neurotransmitters. The postsynaptic receptors on the muscle detect the transmitter which results in a depolarization of the muscle fiber. There are various possibilities for the variation in quantal responses. Some of those possibilities involve the location where neurotransmitter is released and such variation would result in two or more groups of functions appearing in the data. Thus, finding groups of functions in the data may indicate multiple sites active in the synapse.

In searching for different sites, we wish to find groups of functions within the 61 functions shown in Figure 1a. This is our motivation for fitting a mixture of shape invariant functions, where the different underlying shape functions may indicate different active zones. Previous work in this area has uncovered groupings based on functionals of the traces, for example by fitting a normal mixture model on the peak amplitudes [12]. Of course, individual functionals only utilize a small amount of the information available in the entire voltage function. It would also be possible to fit a single shape function g and then fit a mixture model over the parameters (a, b, c, d) , which we do not pursue here.

A brief outline of this paper is as follows. In the next section we introduce MSIMs. We describe our computational techniques, including necessary adjustments to the BARS algorithm, in Implementation. In we then describe methods for evaluating the results (this is nontrivial due to identifiability issues present in SIMs in general), increasing MCMC mixing and selecting the number of components in the mixture. In Results we discuss the results of the simulations and the analysis of the data in Figure 1a. Finally, we provide a discussion of the results.

Mixtures of Shape Invariant Models

Prior and likelihood structure

Suppose we observe I individuals and for each individual we observe data pairs (x_{ij}, y_{ij}) for $i = 1, \dots, I$ and $j = 1, \dots, J_i$. We also assume there exist underlying shape functions g_1, \dots, g_K such that the i th individual follows shape g_k with probability p_k , where $p = (p_1, \dots, p_K)$ is a vector of probabilities. Let z_i be a group indicator where $z_i = k$ indicates that the i th individual follows the k th function and that $\theta_i = (a_i, b_i, c_i, d_i)$ are the self-modeling coefficients for the i th individual. Finally σ^2 is the error variance. Where possible, we place conditionally conjugate priors on the parameters to ensure an easier MCMC implementation. In what follows g refers to the collection of g_1, \dots, g_K and θ refers to the collection of $\theta_1, \dots, \theta_I$, and $N(\mu, \sigma^2)(x)$ refers to a normal density with mean μ and variance σ^2 evaluated at x , with similar notation for other densities. The general prior and likelihood structure is below. We will describe the priors on the θ_i and g_k separately.

$$\pi(\theta, g, \sigma^2, p) = \left[\prod_{i=1}^I \pi_{\theta_i} \right] \left[\prod_{k=1}^K \text{BARS}(g_k) \right] \pi(\sigma^2) \pi(p)$$

$$\pi(\sigma^2) = I \text{nvGamma}(\alpha_\sigma, \beta_\sigma)(\sigma^2)$$

$$\pi(p) = \text{Dir}(\lambda_1, \dots, \lambda_K)(p_1, \dots, p_K)$$

$$z_i | \theta, g, \sigma^2, p \sim \text{Multinomial}(1, p_1, \dots, p_K)$$

$$f_i(x) = a_i g_{z_i}(c_i x + d_i) + b_i$$

$$Y_{ij} | x_{ij}, z_i, \theta, g, \sigma^2, p \sim N(f_i(x_{ij}), \sigma^2)$$

Our inferential goals are to estimate the underlying shape functions, g_1, \dots, g_K , the self-modeling coefficients $(a_i, b_i, c_i, d_i) = \theta_i$ for each individual function f_i , the posterior probabilities of each individual function belonging to each group, and common error variance σ^2 .

Priors on shape functions, g_k

We utilize BARS to estimate each of the g_k functions. Each g_k has a B-spline basis expansion

$$g_k(x) = \sum_{r=1}^{R_k+3} \beta_{kr} b_{kr}(x),$$

where b_1, \dots, b_{R_k+3} form a B-spline basis with knots ξ_1, \dots, ξ_{R_k} and spline coefficient vector β . BARS utilizes a reversible jump MCMC (commonly referred to as RJMCMC) algorithm which samples over the knots in a B-spline used to estimate the function of interest. Without loss of generality, assume the function of interest (for us one of the g_k functions) is supported over the unit interval $(0, 1)$. A B-spline basis is formed using a vector of knots. The prior on this vector assumes the number of knots follows a distribution and that the individual knots are uniformly distributed throughout $(0, 1)$. BARS proceeds by iteratively adding, removing, and relocating knots. These changes in the knot structure are accepted or rejected according to BIC (BIC uses the maximum likelihood estimates of the spline coefficients). As BIC is used, this is very close asymptotically to using a unit-information prior [13] on the spline coefficients β . This structure is what is intended by the notation $\text{BARS}(g_k)$ used above.

Priors on the self-modeling coefficients

Finally, for estimating the self-modeling coefficients $\theta_i = (a_i, b_i, c_i, d_i)$, we place a vague bivariate normal prior on a_i and b_i (these parameters act as regression coefficients). In general, there is no conditionally conjugate prior for c_i and d_i . For functions with a single dominant peak, a transformation of c_i and d_i to a location-scale family is useful.

Suppose for example the underlying shape function g has a peak at x_g^* . The individual function f_i would have that peak at the x where $c_i x + d_i = x_g^*$ which is $x = (x_g^* - d_i) / c_i$. Note that changes in either c_i or d_i result in changes to the peak location of f_i . As proper alignment of the peak is very important to achieving a high likelihood, this induces a correlation in the posterior distribution of c_i and d_i . We have found better results (in terms of the mixing properties of the MCMC chain) by transforming to c_i (which controls the "spread" of the curve) and $m_i = (x_g^* - d_i) / c_i$, which represents the location of the peak. Thus, instead of $f_i(x) = a_i g(c_i x + d_i) + b_i$, we use the transformation $f_i(x) = a_i g(c_i(x - m_i) + x_g^*) + b_i$. For a more thorough description of curve registration techniques and other, potentially helpful transformations for improving MCMC mixing, we refer the reader to the original paper by Ramsay and Li [14] and Gilks and Roberts [15], respectively.

To see the benefit of this transformation by example, let g be a triangular "tent function" where $g(x) = 0$ outside $(0.2, 0.6)$, $g(x)$ increases linearly from 0 to 1 over $(0.2, 0.4)$ and decreases linearly from 1 to 0 over $(0.4, 0.6)$. We generated data using $f(x) = g(0.5x - 0.25)$ for $x = (0, 0.01, 0.02, \dots, 2)$ and $N(0, 1)$ error. The peak of g occurs at 0.4 and the corresponding peak of f occurs at $m = 1.3$. Keeping everything fixed at

the true value except c and d , Figure 2a shows the loglikelihood over (c, d) . As can be seen, there is a distinct correlation between c and d which limits the mixing of the MCMC chain. In contrast, Figure 2b shows the loglikelihood as a function of c and m , which provides a much better structure for MCMC.

Of course, this transformation is limited to g with a single peak, which is not always true. Thus, this transformation will not produce an improvement in general, or other transformations may be useful in other contexts. For our synaptic transmission data, we have found this transformation to be quite useful. Thus, for data with a dominant peak:

$$\pi = (a_i, b_i, c_i, m_i) = [N_2(\mu_{a,b}, \Sigma_{a,b})(a_i, b_i)] [Gamma(\alpha_c, \beta_c)(c_i)] [U(0,1)(m_i)]$$

Identifiability

Identifiability is a pervasive issue in self-modeling regressions [2,4]. Suppose for example we have a particular underlying shape function $g(x)$ and a set of coefficients (a, b, c, d) . One could achieve the same f by taking $g^*(x) = 2g(x)$ and taking $a^* = a/2$. Generally, one can take any self-modeling transformation of g and then adjust (a, b, c, d) appropriately to produce the same f . In the original Lawton paper (1), this problem was addressed by forcing g to have particular properties (such as a maximum of 1 and minimum of 0). However, in the Bayesian formulation such constraints would require any estimation method to be significantly altered. BARS, which we utilize here, is not obviously adjustable to handle constraints on the function.

An alternative method of forcing identifiability is to pick an individual function f_i and force its corresponding (a, b, c, d) to be $(1, 0, 1, 0)$. This method is utilized in previous work [10]. This forces the scaling on g to follow the chosen function but allows BARS to run unconstrained.

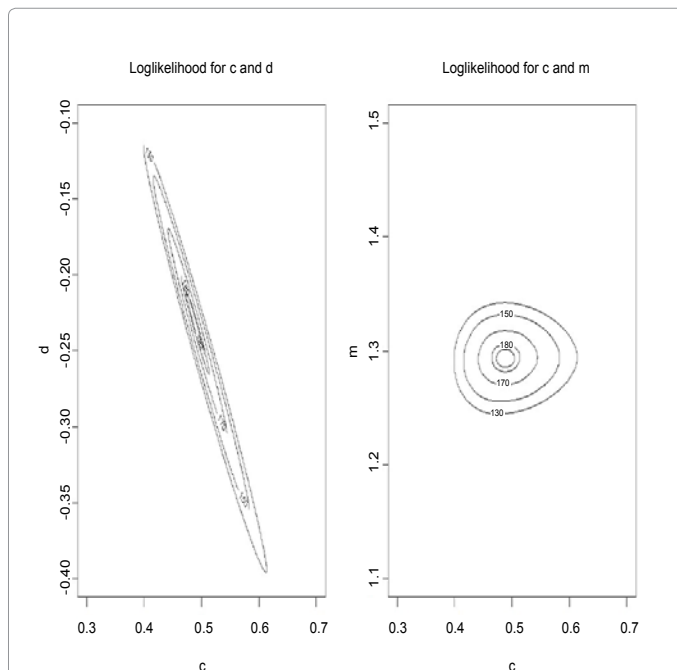


Figure 2: Loglikelihood structures of shift-scale and location-scale self-modeling coefficients. The left pane (a) shows the structure for the (c, d) self-modeling coefficients, compared to the right pane (b) which shows the (c, m) location-scale transformation. For a function with a single dominant peak, the location-scale transformation has a much lower correlation between c and m and thus the corresponding MCMC algorithm has superior mixing properties.

The central theme here is that in the self-modeling paradigm you are really estimating relative shifts and scalings among the functions.

Unfortunately, in the mixture setup this is not feasible, as there are multiple shape functions. Since we do not know a priori which functions, f_1, \dots, f_i belong to which group (e.g. which are self-modeling versions of g_1 , which are self-modeling versions of g_2 , etc.) we cannot select functions to use as "anchors" in each group. Thus the simplest option seems to be to leave a nonidentified model. Altman and Villarreal [4] also use an unconstrained formulation without difficulty, though from a frequentist standpoint.

Note, however, that our central inferential goals are the assignment of functions to groups, the estimation of the individual functions, and the underlying shape of the g_1, \dots, g_K functions. The first two are identifiable, and g_1, \dots, g_K are identifiable up to self modeling versions. As with any mixture model, there is a secondary identifiability problem in that the component labels may be switched, but component label switching [14] has not been an issue in the analyses we performed. Although not addressed in this work, there is a third identifiability problem in mixture models arising from having an unnecessarily large number of mixtures when a smaller number of mixtures is actually sufficient.

Implementation

We implemented the model in Section 2 using an MCMC scheme. Some parameters are of course easier to update than others.

Sampling scheme

Updating σ^2 and p : Both σ^2 and p have straight forward conjugate priors conditional on the rest of the parameters.

$$\pi(p | rest) = Dir(\lambda_1 + \# Z_i = 1), \dots, \lambda_K + (\# Z_i = K) (p_1, \dots, p_K)$$

$$\pi(\sigma^2 | rest) = InvGamma(\alpha_\sigma + \frac{N}{2}, \beta_\sigma + \sum_{i,j} e_{ij}^2 / 2), \quad (1)$$

where

$$e_{ij}^2 = (y_{ij} - \hat{y}_{ij})^2 \quad N = \sum_i J_i$$

Shape functions g_1, \dots, g_K : Conditional on the remaining variables, each g_k is conditionally independent, with each g_k being updated separately using only those data functions with $z_i = k$.

Each g_k is defined by its corresponding knot set ξ_k and spline coefficients β_k . In the original BARS algorithm [7], updating was done entirely on ξ using BIC with the maximum likelihood estimate of β . While we update ξ using the BARS updating, we actually draw $\xi, \beta | rest$ via $\xi | rest$ and $\beta | \xi, rest$. Our reason for drawing β values is that, unlike the original work [7], we have additional parameters (specifically z_i and θ_i) that are easier to simulate if β is fixed.

Updating Z_i : Updating each Z_i also requires moving functions between groups, as there are subject specific parameters θ_i for each variable. In the MCMC scheme one must take into account that (a, b, c, d) have different interpretations depending on which group the function belongs to (e.g. the outcome of Z_i). If g_1 and g_2 have different peaks, for example, then a perfect fit for g_1 would require a much different value of α than a perfect fit for g_2 .

Ideally, we would like to avoid this problem and integrate the self-modeling coefficients out to find

$$Pr(Z_i = k | g_1, \dots, g_K, p, \sigma^2) = \int Pr(Z_i = k | g_1, \dots, g_K, p, \sigma^2, \theta_i) \pi(\theta_i) d\theta_i$$

Unfortunately, it is not computationally efficient to find this

integral nor is it efficient to approximate it (for example finding the MLE of θ_i would require numerical maximization). Thus in our sampler we utilize θ_{ik} (different θ_i for each k). To draw a new Z_i , note

$$Pr(Z_i = k | g_1, \dots, g_K, p, \sigma^2, \theta) = \frac{p_k \prod_j N(f_{ik}(x_{ij}, \sigma^2)(y_{ij}))}{\sum_{w=1}^K p_w \prod_j N(f_{iw}(x_{ij}, \sigma^2)(y_{ij}))}$$

where

$$f_{ik}(x_{ij}) = a_{ik}g_{ik}(c_{ik}x_{ij} + d_{ik}) + b_{ik}$$

Updating the self-modeling coefficients: Finally, the self-modeling coefficients (a, b) (recall in the sampler we have separate self-modeling coefficients θ_{ik} for each i and k) are updated straightforwardly, as they are regression parameters with conjugate priors. The (c, m) coefficients are updated using a Metropolis-Hastings random walk scheme.

Singularities in the BARS updating

In the original BARS implementation, proposed knot sets which resulted in singular design matrices (loosely, too many knots in an area with too few data points), were discarded [16,17]; these proposals may be viewed simply as a data dependent prior. In the self-modeling setup, there is an additional complication. In one iteration we may achieve a nonsingular design matrix, but as the self-modeling coefficients θ_i are adjusted, this moves m_i defined in Section 2.1.2 and may result in a singularity in the design matrix. We additionally reject these moves out of hand in the sampler. At present, these rejections in the sampler are not a frequent occurrence, but seems an unavoidable problem as long as BIC is used in the BARS steps.

Starting values

To initialize the chain, all f_i functions were first aligned to the largest function (the trace with the largest difference between the maximum and minimum values) using a Procrustes registration [14]. We then computed the mean square error (MSE) between each aligned trace and the largest trace. Focusing on a two-component mixture, the half of the traces with the smallest MSE values were placed in one group while the half of the traces with the largest MSE values were placed in the second group. Essentially, this means the half of the traces closest in shape (not necessarily scale) to the largest trace were placed together in a group. This initialized the Z_i values. To initialize g_1 and g_2 in the two component mixture model, we fit a spline to the traces in each group (we simply assume 9 equally spaced knots initially, which is then adjusted by the sampler). The initial spline coefficients are then computed by aligning each individual trace to its corresponding g_{Z_i} and finally the error variance is computed by taking the MSE of the entire fit.

Assessing Implementation Results

Posterior inference

At the end of the MCMC run we have a set of M iterations. For each $m=1, \dots, M$, the m^{th} iteration contains functions $g_1^{(m)}, \dots, g_K^{(m)}$ (expressed in a spline basis), mixing proportions $p^{(m)}$, an error standard deviation $\sigma^{(m)}$ and component assignments $z_1^{(m)}, \dots, z_I^{(m)}$ for each individual function. Finally, for each i we have self-modeling coefficients $\theta_i^{(m)}$. We assess the results in terms of the identifiable pieces of the sampler.

1. We directly have the mixing proportions p between the different functions g_k . We also compute the estimated posterior probability each individual trace belongs to each group by finding the observed

proportion of times $z_i^{(m)} = k$ for each k . Thus, we acquire overall proportions of functions in each group and the estimated posterior probabilities each individual function belongs to each individual g_k .

2. To assess the fit of the individual functions, we computed the estimated $f_i(t)$ by constructing for each iteration and each trace

$$\hat{f}_i^{(m)}(t) = a_i^{(m)}g_{z_i}^{(m)}(c_i^{(m)}t + d_i^{(m)}) + b_i^{(m)}$$

$$\bar{f}_i(t) = \frac{1}{M} \sum_{m=1}^M \hat{f}_i^{(m)}(t) \quad (2)$$

In the plots that are constructed from the simulation studies and EPSPs application in Section 5, we plot a selection of the $\hat{f}_i^{(m)}(t)$ functions (iterations $m = 100, 200, \dots$) and the overall average $\bar{f}_i(t)$.

To estimate each g_k we must remember that each g_k is allowed to drift in the sampler, and thus the resulting $g_k^{(m)}$ iterations must be aligned in shape before producing an estimate. Thus, we begin by taking the functions $g_k^{(m)}$ across the iterations and aligning them to each other. This was done via a Procrustes registration as in previous work [10]. The base shape function of this alignment was used as the estimate for g_k .

The amount of burn in necessary for Gibbs sampling in the MSIM setting depends on data-specific and method-specific factors. Wallstrom et al. [17] recommend up to 20,000 burn-in iterations for the BARS implementation with normally-distributed data. We found that additional iterations were necessary to achieve proper burn in for the self-modeling coefficients.

Improving acceptance rates

It is important that the sampling mechanism provide a sufficient amount of switching in order to explore the state space well. Typically, one would choose a reversible-jump mechanism to facilitate this switching, but in most RJMCMC scenarios it is easier to sample from the posterior distribution within each subgroup (i.e. mixture component). The aforementioned sampling mechanism in Section 3 uses metropolis random walks within each subgroup, which may lead to lower acceptance rates, thereby lowering the degree of switching between groups.

To facilitate switching between groups, we propose using the approximating posterior distributions of the self-modeling coefficients in θ_i from our previous MCMC implementation in Section 3 to conduct a second MCMC. In this second MCMC, acceptance rates may be improved as follows. Let $\alpha_{i,k \rightarrow k'}$ be the probability of the i th trace moving from the current state M_k to the candidate state $M_{k'}$. For ease of description, we will now omit notation for trace i . The acceptance probability for moving from the current state (k) to the candidate state (k') is

$$\alpha_{k \rightarrow k'} = \frac{\pi(M_{k'})\pi(a_k, b_k, c_k, d_k | M_{k'})\pi(y | a_k, b_k, c_k, d_k, M_{k'})}{\pi(M_k)\pi(a_k, b_k, c_k, d_k | M_k)\pi(y | a_k, b_k, c_k, d_k, M_k)} \times \frac{p_{k'} \times f_k(a_k, b_k, c_k, d_k)}{p_k \times f_{k'}(a_k, b_k, c_k, d_k)} \quad (3)$$

where in general notation for state m , $\pi(M_m)$ is the prior probability that the trace has underlying shape function g_m ; $f_m()$ refers to coefficients (a_m, b_m, c_m, d_m) being evaluated at the multivariate normal distribution with respective mean vector p_m and covariance matrix Σ_m obtained from the complete conditional distributions in the first MCMC implementation; p_m is the empirical probability that the trace belonged with shape function $g_m()$ in the first MCMC; $\pi(y | a_m, b_m, c_m, d_m, M_m)$ is the likelihood for the trace under shape function g_m . The $\pi(a_m, b_m, c_m, d_m | M_m)$ corresponds to coefficients (a_m, b_m, c_m, d_m)

being evaluated using priors under M_m , which can be simplified as $\pi(a_m, b_m, c_m, d_m | M_m) = \pi(a_m, b_m, c_m, d_m | M_m) \pi(c_m | M_m) \pi(d_m | M_m)$; prior $\pi(d_m | M_m)$ is an induced prior that we compute via transformations. The algorithm for computing the acceptance probabilities is outlined in Supplementary Information.

Determining the number of mixtures

One outstanding issue still to be addressed is inferentially determining the number of groups present in the data. When making this determination, BIC is intuitively a reasonable choice; however, regularity conditions do not hold in the mixture model setting [18]. Furthermore, BIC is not reliable in the mixture model settings with smaller sample sizes. Pilla and Charnigo [19] proposed a flexible information criterion (FLIC) specific to selecting the number of components in semiparametric mixture models, which performs better than BIC in settings where the mixture components are not easily distinguishable and better than AIC when sample sizes are large. We adapted the FLIC implementation used previously for fitting a normal mixture model to a distribution of birth weight [20] as follows. For a mixture of self-modeling regressions with components $k=1, \dots, K$

$$FLIC_K = -2 \log L_K + 2(\log \sqrt{N})^{B(N, \delta)} \nu, \quad (4)$$

where, L_K is the log likelihood of the data evaluated at the posterior mean estimates for each function f_i , $i=1, \dots, I$; N is the total number of observations defined in Equation (1). The number of free parameters in the model (ν) is the number of shape functions g_k (note that we do not consider the number of parameters in the knot and spline coefficient vectors since we can use the posterior mean \bar{g}_k from the sampler, post drift alignment, as our estimate for g_k), plus the number of undetermined mixing proportions ($K-1$ free parameters), the total number of self-modeling coefficients in the 4×1 vector θ_i for all I subjects, and the common variance parameter σ^2 . The bivariate ratio function $B(N, \delta)$ and penalty statistic δ have been defined previously [19]. For each MSIM considered, we can compute the fraction of between-component variability to within-component variability across the x values (or time points) and use these results to compute the average fraction of between-component variability to within-component variability (see formula in Supplementary Information).

Results

The MCMC implementation and post processing described in Sections 3-4 were executed for simulation studies and the motivating EPSPs example using R [21].

Simulations

We simulated data to identify how well the algorithm discriminated two separate, but similar functions. In each sample dataset we used the following underlying shape functions:

$$g_1(x) = 0.2 \exp\{-20(x-0.2)^2\}$$

$$g_2(x) = (\exp\{-0.7x\} - \exp\{-12x\})I_{(0,1)}$$

These functions are similar to some parametric models used in the biology literature for EPSPs, and are quite similar in shape. Figure 3 shows the two underlying functions. The two functions are similar in shape, but they are clearly not identical. In other words, these two shape functions are clearly not self-modeling versions of each other. Although g_1 is defined as a symmetric function, its alignment with g_2 makes it appear asymmetric over the range of x . The function g_2 is not differentiable at $x=0$ (note self-modeling versions with a slight shift are

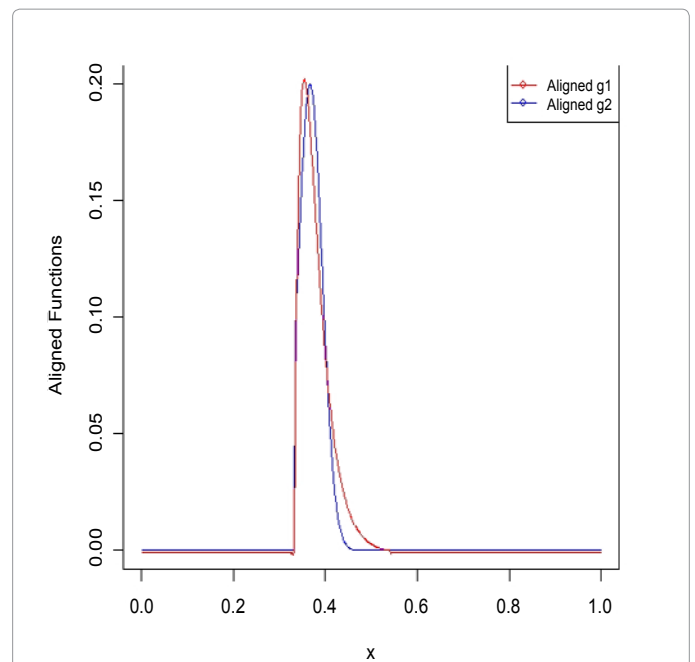


Figure 3: Shape functions used for simulations. These two functions are aligned versions of $g_1(x)=0.2\exp\{-20(x-0.2)^2\}$ and $g_2(x)=(\exp\{-0.7x\}-\exp\{-12x\})I_{(0,1)}$. They are similar, but certainly NOT identical, in shape. Note also g_2 is not differentiable at the initial rise, which creates a situation where multiple knots are desirable in that region. The simulations consisted of generating 40 datasets, each with 100 functions, 50 of which are self-modeling versions of g_1 with error and 50 of which are self-modeling versions of g_2 with error.

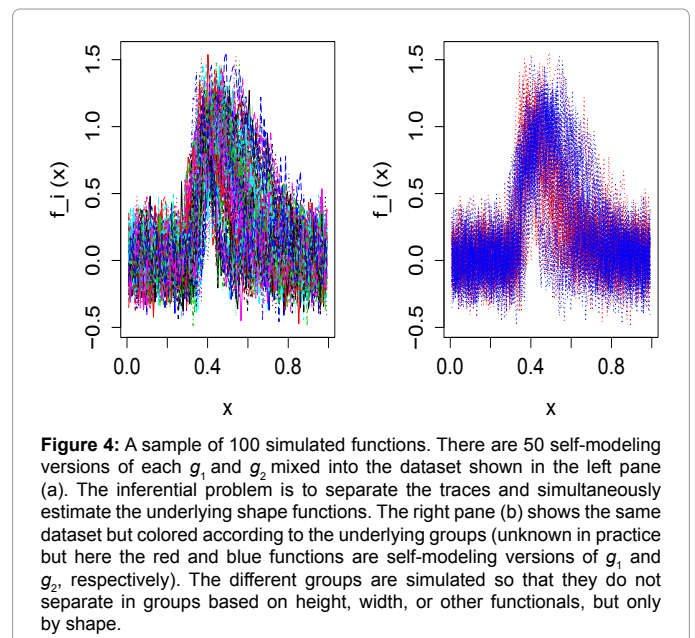


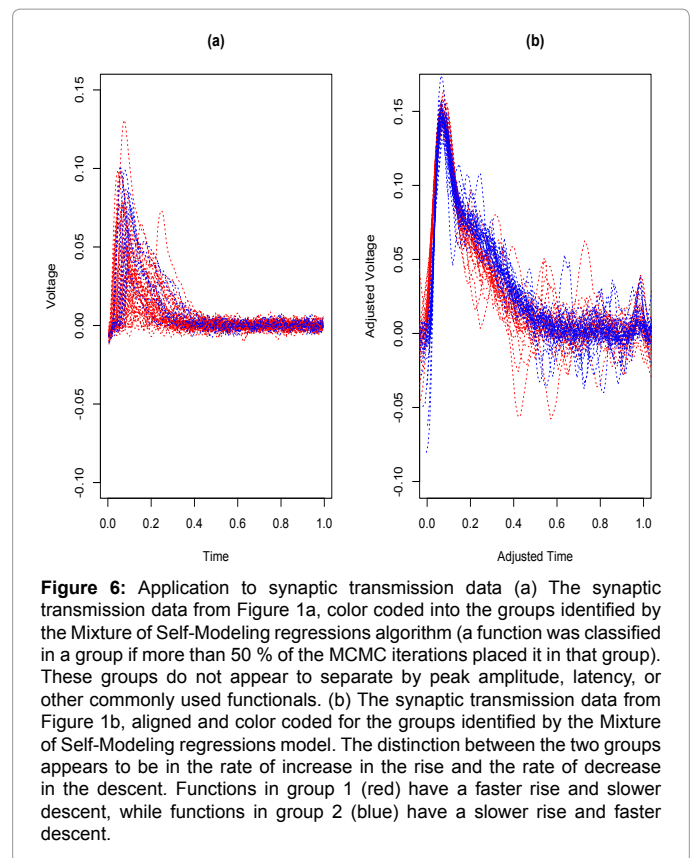
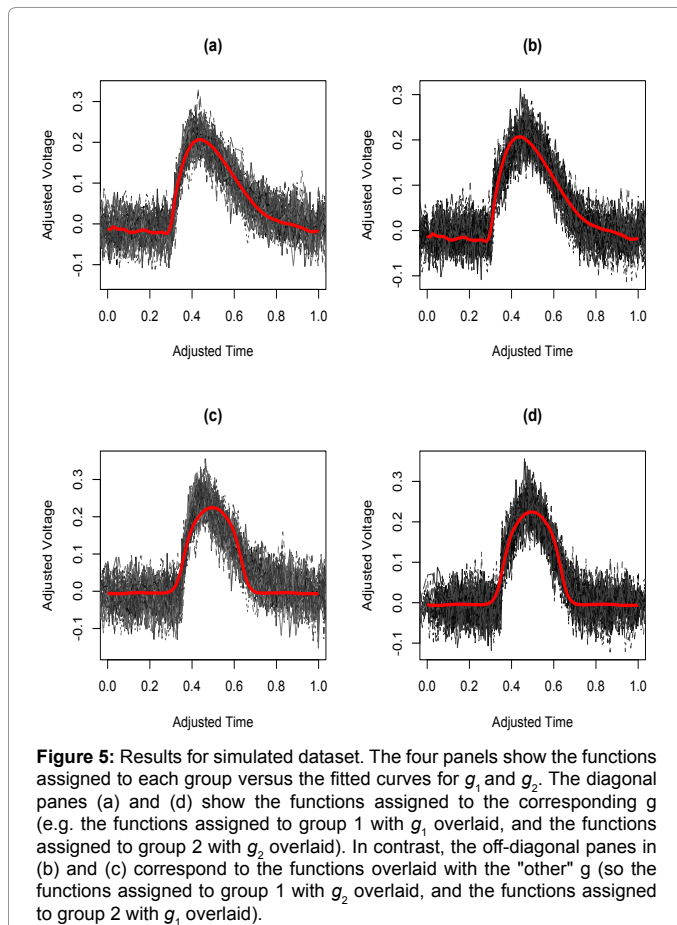
Figure 4: A sample of 100 simulated functions. There are 50 self-modeling versions of each g_1 and g_2 mixed into the dataset shown in the left pane (a). The inferential problem is to separate the traces and simultaneously estimate the underlying shape functions. The right pane (b) shows the same dataset but colored according to the underlying groups (unknown in practice but here the red and blue functions are self-modeling versions of g_1 and g_2 , respectively). The different groups are simulated so that they do not separate in groups based on height, width, or other functionals, but only by shape.

quite similar, so this cusp could appear at a different x value in the estimate, see the discussion of identifiability in Section 2.2), which creates a point where multiple knots are desirable. This is a situation where adaptive knot splines are more effective than using known knots.

Each sample dataset consisted of 100 functions, 50 generated from

self-modeling versions of g_1 and 50 from self-modeling versions of g_2 . The self-modeling coefficients (a, b, c, d) were randomly chosen for each function in each simulation to provide a range of overlapping functions. The range of values from which a, b, c, d were selected consisted of (3.0,7.0), (-0.15,0.10), (0.15,0.90), and (-0.06,0.80), respectively. The x values for each function correspond to 121 equally-spaced points in the interval (0,1). Forty simulated datasets of 100 functions were generated, 20 of which had an error standard deviation of $\sigma=0.15$ while the remaining 20 had an error standard deviation of $\sigma=0.05$. One sample dataset (for $\sigma=0.05$) is shown in Figure 4a. Figure 4b shows the same dataset color coded by the appropriate γ function. It is certainly not obvious from the data alone which γ is appropriate. The first- and second-stage MCMC algorithms from Equation (3) were run on each simulated dataset. The first-stage sampler was run for 80,000 iterations with the first 20,000 removed as burnin (this is perhaps excessive, but burn-in is long in this context and we were quite conservative). The second sampler was run for 30,000 iterations with the first 10,000 iterations removed. For each simulated dataset we computed the assessable quantities described in Section 4 (g_1 and g_2 estimated up to shape invariance, the individual functions f_p , the probabilities of correct classification and error variance σ^2).

Figure 5 shows the estimated g_1 and g_2 plotted against the group assignments (for plotting, each function was assigned to the group that functions spent the most time assigned to in the second MCMC run) for the simulated dataset from Figure 4. The diagonal panes in Figure 5 show the functions (aligned) assigned to group 1 plotted against the fitted g_1 and the functions assigned to group 2 (aligned) plotted against



the fitted g_2 . The off-diagonal panes show each g_k plotted against the functions which were NOT assigned to that group.

Overall, correct classification rates for the functions were quite high and had similar rates across the simulations for both the first and second MCMC runs. Of the 20 datasets generated with lower noise $\sigma=0.05$, all but 3 had all 100 functions classified correctly. For those 3 datasets, 2 had 99 functions correctly classified and the other had 97 functions correctly classified. For the 20 datasets with $\sigma=0.15$, there was more variability because of the increased noise but there were still high correct classification rates, ranging from a low of 77 classified correctly to a high of 94.

An improvement in switching was defined as moving from the current state to the candidate state more often during post-burnin iterations in the second MCMC, compared to movement in the first MCMC. For higher noise ($\sigma=0.15$) data, all functions had improved acceptance rates. In lower noise ($\sigma=0.05$) data, switching improved in all but 3 datasets. For those 3 datasets, there were 4, 7, and 25 functions that did not improve their acceptance rates in the second run.

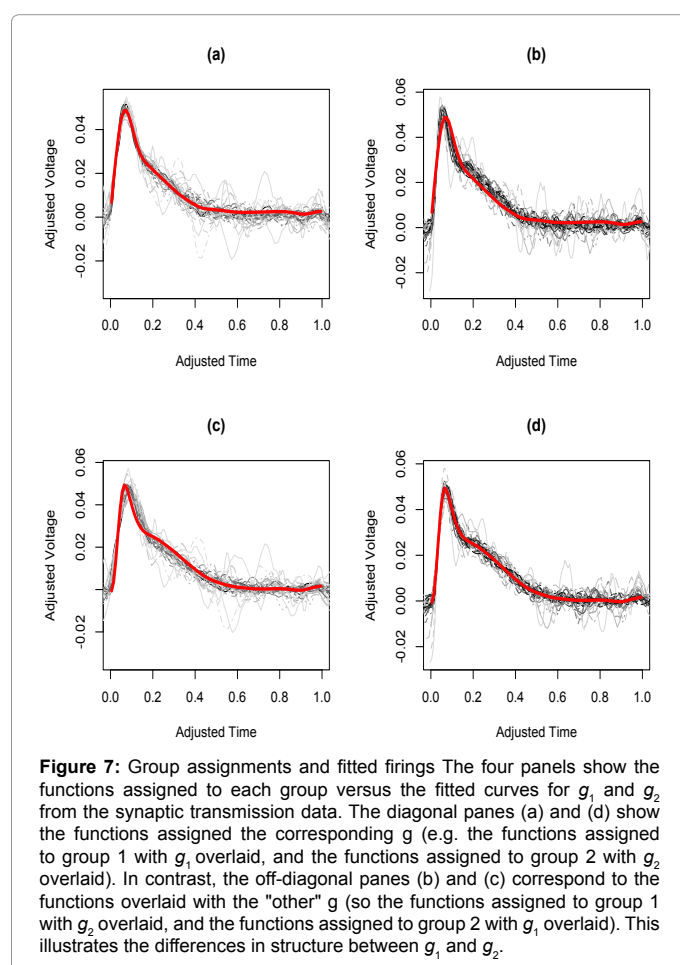
We calculated FLIC from Equation [4] for each simulation using one-group and two-group results. All lower-noise simulations favored including two distinct shape functions g_1 and g_2 over a single shape g . Eighteen of the higher-noise datasets favored the two-group model while 2 of the datasets had FLIC results favoring the single group model.

Classifying firings from synaptic transmission data

Of the 61 firings, 31 were classified to group 1 over 50% of the time while the other 30 were classified to group 2 over 50% of the time (to make sure the algorithm simply didn't split the 61 functions in half

arbitrarily, we also fit subsets of the functions and found consistent assignments). Figure 6a is identical to Figure 1a except the functions are color coded into group 1 (red) and group 2 (blue). Acceptance probabilities were increased in the second MCMC run, but group membership changed for only 2 of the 61 firings from the first run to the second run.

There is no obvious pattern to the assignments in Figure 6a (e.g. group 1 traces are not generally higher, wider, etc.). When we construct the plot analogous to the aligned data in Figure 1b, we do see a pattern emerge. In Figure 6b, the central difference in the groups appears near the aligned peaks. Group 1 (in red) shows a quick increase, followed by a slower decrease after the peak. In contrast, group 2 (in blue) has a slower increase to the peak, but a quicker decrease from the peak. Figure 7 shows the individual estimates of g_1 and g_2 plotted against the functions assigned (here meaning greater than 0.5 posterior probability) to each group. Thus the diagonal panels in this figure correspond to good fits (e.g. the functions assigned to group 1 with g_1 overlaid, and the functions assigned to group 2 with g_2 overlaid); the off diagonal panels show the differences between the groups (e.g. the function assigned to group 1 with g_2 overlaid, and the functions assigned to group 2 with g_1 overlaid). We compared the model results from the FLIC for a single shape versus two shapes and found the model with two shapes was more favorable. Again, the two different shapes are apparent in the graph.



Discussion

Fitting an MSIM allows for multiple shapes to be present in a sample of functions. In substantive contexts like the synaptic transmission data, finding these groups has a direct impact on questions like the number of active zones in a synapse. The algorithm proposed in this paper behaves well in simulations and provides results for the synaptic transmission data consistent with biological expectations.

Recent work by Zhang and Telesca [22] focuses on clustering and general registration of functional data where the number of knots and their locations are specified a priori and common to all underlying shapes. They show this approach works well for clustering growth data and gene expression curves. Our application to EPSPs and simulation studies address settings where the underlying shape functions are quite similar but perhaps one or more of the shape functions are not differentiable over the entire interval of interest. In these settings, we have found it advantageous to use BARS, compared to using known knots, as adaptive methods can include multiple knots to estimate cusps or other sharp changes in curvature.

The degree of switching (i.e. changes in acceptance rates) generally increased from the first MCMC to the second MCMC but appeared to depend on overall noise in the data. These results are consistent with existing literature on tuning proposal distributions [23]. It is possible that other methods of "data-driven" tuning which incorporate overall variance in the MSIM may be necessary to further improve acceptance rates in the RJMCMC scheme [24].

Substantively, the finding of groups is expected, as such a finding can be directly linked to the underlying structural entities which are responsible for efficacy in chemical synaptic transmission in general [12,25,26]. However, if one fits a model with two groups, then two groups will appear in the results. While we were able to adapt an existing information criterion for this assessment [19] and obtain reasonable results, in the future it will be important to formally develop an inferential procedure for MSIMs. Fortunately for the synaptic transmission data, simpler mixture models have also detected two groups [12].

In univariate Gaussian mixture models, there are various model selection criteria that could be readily supplied [18,27] or one could use reversible-jump samplers to move between models with differing numbers of components in a manner similar to a previously-described approach [28]. In our context it is unclear how to implement a model selection criteria that requires the number of parameters in the model, as the spline formulation includes the number of knots as one of the unknown parameters. In this work, we substitute these unknown parameters with the K parameters corresponding to estimating \bar{g}_k for $k=1,\dots,K$. Even simplistic criteria such as investigating the MSE between functions, or more precisely the use of χ^2 or noncentral χ^2 distributions, is complicated by the differing scaling produced by the self-modeling coefficients for each function. It would be of interest in future work to formally incorporate spline complexity in the fit statistic evaluation.

Acknowledgement

This research was supported by NCRR(NIH) Grant P20 RR16481. The authors are grateful to Richard Charnigo, PhD, for sharing implementation code for the flexible information criterion. The authors thank the reviewers for their comments, which have substantially improved the paper.

References

1. Lawton W, Sylvestre E, Maggio M (1972) Self-Modeling Nonlinear Regression. *Technometrics* 14: 513-532.

2. Kneip A, Gasser T (1988) Convergence and Consistency Results for Self-Modeling Nonlinear Regression. *Annals of Statistics* 16: 82-112.
3. Telesca D, Inoue L (2008) Bayesian Hierarchical Curve Registration. *Journal of the American Statistical Association* 103: 328-339.
4. Altman N, Villarreal J (2004) Self Modeling Regression with Random Effects Using Penalized Splines. *Canadian Journal of Statistics* 32: 251-268.
5. Lindstrom ML, Bates DM (1990) Nonlinear mixed effects models for repeated measures data. *Biometrics* 46: 673-687.
6. Cole TJ, Donaldson MD, Ben-Shlomo Y (2010) SITAR—a useful instrument for growth curve analysis. *Int J Epidemiol* 39: 1558-1566.
7. DiMatteo I, Genovese C, Kass R (2001) Bayesian Curve Fitting with Free Knot Splines. *Biometrika* 88: 1055-1071.
8. Behseta S, Kass RE (2005) Testing equality of two functions using BARS. *Stat Med* 24: 3523-3534.
9. Behseta S, Kass R, Wallstrom G (2005b) Hierarchical models for assessing variability among functions. *Biometrika* 92: 419-434.
10. Viele K, Lancaster M, Cooper RL (2006) Self-modeling structure of evoked postsynaptic potentials. *Synapse* 60: 32-44.
11. Lancaster M, Viele K, Johnstone AF, Cooper RL (2007) Automated classification of evoked quantal events. *J Neurosci Methods* 159: 325-336.
12. Viele K, Stromberg AJ, Cooper RL (2003) Estimating the number of release sites and probability of firing within the nerve terminal by statistical analysis of synaptic charge. *Synapse* 47: 15-25.
13. Kass R, Wasserman L (1995) A Reference Bayesian Test for Nested Hypotheses and its Relationship to the Schwarz Criterion. *Journal of the American Statistical Association* 90: 928-934.
14. Ramsay J, Li X (1998) Curve Registration. *Journal of the Royal Statistical Society, Series B* 60: 351-363.
15. Gilks WR, Roberts GO (1996) Chapter 6: Strategies for improving MCMC. *Markov Chain Monte Carlo in Practice*. (1stedn), Chapman and Hall/CRC Press.
16. Jasra A, Holmes CC, Stephens DA (2005) Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modeling. *Statistical Science* 20: 50-67.
17. Wallstrom G, Liebner J, Kass RE (2008) An Implementation of Bayesian Adaptive Regression Splines (BARS) in C with S and R Wrappers. *J Stat Softw* 26: 1-21.
18. Steele R, Raftery A (2009) Performance of Bayesian Model Selection Criteria for Gaussian Mixture Models. Technical Report 559, Department of Statistics, University of Washington.
19. Pilla R, Charnigo R (submitted) Consistent estimation and model selection in semiparametric mixtures.
20. Charnigo R, Chesnut L, LoBianco T, Kirby RS (2010) Thinking outside the curve, part I: modeling birthweight distribution. *BMC Pregnancy and Childbirth* 10: 37.
21. R Core Team (2013) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
22. Zhang Y, Telesca D (2014) Joint Clustering and Registration of Functional Data.
23. Roberts GO, Gelman A, Gilks WR (1997) Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms. *The Annals of Applied Probability* 7: 110-120.
24. Rosenthal JS (2011) Chapter 4: Optimal proposal distributions and adaptive MCMC. *Handbook of Markov Chain Monte Carlo*. (1stedn), Chapman and Hall/CRC Press.
25. Cooper RL, Stewart BA, Wojtowicz JM, Wang S, Atwood HL (1995) Quantal measurement and analysis methods compared for crayfish and *Drosophila* neuromuscular junctions, and rat hippocampus. *J Neurosci Methods* 61: 67-78.
26. Cooper R, Marin L, Atwood H (1995b) Synaptic Differentiation of a Single Motor Neuron: Conjoint Definition of Transmitter Release, Presynaptic Calcium Signals, and Ultrastructure. *J Neurosci* 15: 4209-4222.
27. Charnigo R, Sun J (2004) Testing for Homogeneity in Mixture Distributions via the L2 Distance Between Competing Models. *Journal of the American Statistical Association* 99: 488-498.
28. Richardson S, Green P (1997) On Bayesian Analysis of Mixtures with an Unknown Number of Components. *Journal of the Royal Statistical Society, Series B* 59: 731-792.