

Artificial Neural Network Modification of Simulation-Based Fitting: Application to a Protein–Lipid System

Petr V. Nazarov,^{*,†,‡} Vladimir V. Apanasovich,[‡] Vladimir M. Lutkovski,[‡] Mikalai M. Yatskou,[‡]
Rob B. M. Koehorst,[†] and Marcus A. Hemminga[†]

Laboratory of Biophysics, Wageningen University, Dreijenlaan 3, 6703 HA, Wageningen, The Netherlands,
and Department of Systems Analysis, Faculty of Radio Physics, Belarusian State University,
Skaryna Avenue 4, Minsk 220050, Belarus

Received July 21, 2003

Simulation-based fitting has been applied to data analysis and parameter determination of complex experimental systems in many areas of chemistry and biophysics. However, this method is limited because of the time costs of the calculations. In this paper it is proposed to approximate and substitute a simulation model by an artificial neural network during the fitting procedure. Such a substitution significantly speeds up the parameter determination. This approach is tested on a model of fluorescence resonance energy transfer (FRET) within a system of site-directed fluorescence labeled M13 major coat protein mutants incorporated into a lipid bilayer. It is demonstrated that in our case the application of a trained artificial neural network for the substitution of the simulation model results in a significant gain in computing time by a factor of 5×10^4 . Moreover, an artificial neural network produces a smooth approximation of the noisy results of a stochastic simulation.

1. INTRODUCTION

Simulation-based fitting (SBF) has recently become a standard tool for the analysis of experimental data to extract the real parameters of, for example, chemical¹ and biophysical systems.^{2–5} The idea of SBF is the approximation of experimental data by synthetic data obtained via simulation modeling. In comparison to standard analytical data fitting techniques, SBF has the advantage that it fits natural physical and chemical parameters of the system itself and gives direct insight into how they affect the experimental characteristics of the system.²

However in practice SBF has several limitations. The most crucial problem is that simulation modeling usually is a very time-consuming operation, which results in a long fitting time. In some cases this approach is not useful at all, because the time of optimization becomes nonrealistic (from months to years). The origin of another weak point often lies in the stochastic nature of both simulation and experimental data. This results in a very complex behavior of the discrepancy function and the introduction of a large number of local minima.⁶

The aim of our study is to develop and present solutions for these problems. Here we propose to use an artificial neural network^{7,8} (ANN) to speed up the parameter identification and to make the process of fitting less stochastic. The main idea of the method is the substitution of a simulation model by an ANN (specifically a multilayer perceptron^{7,8}) during fitting. Because of the simplicity of the multilayer perceptron structure and the internal math-

ematics, the computation time needed for the calculation of neural network outputs is much less than for simulation modeling. Hence, the replacement of a simulation model by a multilayer perceptron leads to a considerable speeding up of all calculations.

The proposed approach of the neural network approximation was tested on a simulation model of resonance energy transfer⁹ between fluorescent labels of bacteriophage M13 major coat proteins incorporated into a lipid bilayer.

2. THEORY

2.1. Principles of SBF. The SBF approach was developed for the determination of physical and chemical parameters of complex systems, which cannot be completely described by analytical expressions. Let us consider the idea of SBF on the following general example. A complex (physical or chemical) system Θ can be characterized by a vector of parameters $P = (p_1, p_2, p_3, \dots)$. These parameters can be regarded as input parameters of the system Θ . After a number of experimental studies on the system Θ are carried out with different input parameters, the vector of output values F can be obtained. In this case, the system can be considered as an operator performing the following operation:

$$\Theta(p_1, p_2, p_3, \dots) = \Theta(P) = F \quad (1)$$

Usually, some input parameters are known. Let us denote them P_0 ; for example, let $P_0 = (p_1, p_2)$. Other parameters, which should be extracted, are denoted as P_X ; suppose $P_X = (p_3, p_4, \dots)$. The vector of input parameters therefore includes a combination of known and unknown parameters $P = (p_1, p_2, p_3, \dots) = (P_0, P_X)$. The extraction of P_X is the aim of the analysis.

* Corresponding author phone: 31-317-482113; fax: 31-317-482-725; e-mail: petr.nazarov@wur.nl. Address correspondence to the Wageningen University address.

[†] Wageningen University.

[‡] Belarusian State University.

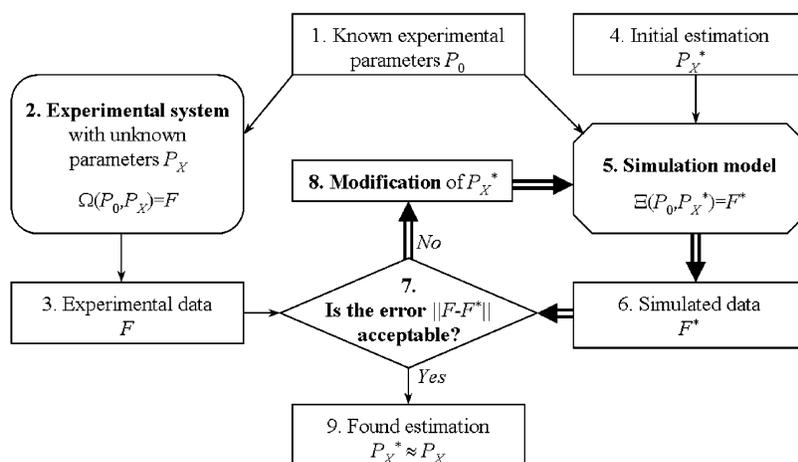


Figure 1. Flow diagram of simulation-based fitting (see text).

Let us assume that for system Θ it is possible to build an adequate simulation model, which performs operation (2) with the same physical parameters P .

$$\Xi(P) \equiv \Xi(P_0, P_X) = F^* \quad (2)$$

where F^* contains the simulated output values, which should approximate the experimental ones.

The determination of the unknown parameters P_X is carried out in the form of SBF. The flow diagram of this method is shown in Figure 1.

The following steps can be identified in SBF:

1. Output values F are obtained experimentally (see blocks 1–3).
2. An adequate model Ξ of system Θ , which performs operation (2), is created (block 5).
3. An initial estimation P_X^* is made for P_X (block 4).
4. An optimization algorithm, using a variation of parameters P_X^* , minimizes the discrepancy function $\|F^* - F\|$ (blocks 6–8 and 5 again).
5. Finally, the fitted parameters P_X^* , which should estimate the experimental parameters P_X , are obtained (block 9).

As was mentioned before, the main problem of SBF is its time expenses. We solve this problem by the application of an ANN, which approximates and substitutes the simulation model during SBF.

2.2. ANN Approximation. As was shown independently by Cybenko¹⁰ and Hornik,¹¹ continuous smooth functions can be uniformly well approximated by linear combinations and superpositions of sigmoid functions, i.e., by a multilayer perceptron. This is the most common class of ANNs.^{8,10–13} Concerning the application of ANNs, three-layer perceptrons have better learning abilities than two layered ones.⁸

Most relationships in chemistry and physics can be represented by continuous functions (if they have a stochastic nature—let us speak about their mean). This gives the possibility of approximating the simulation model Ξ by a multilayer perceptron. Let us denote this approximating ANN transform as Ψ . It performs the operation

$$\Psi(P_0, P_X) = F^{**} \quad (3)$$

where F^{**} is the neural network approximation of the output values of the system.

Hence, instead of the simulation model Ξ , the neural network approximation Ψ can be used during parameter fitting. The suggested ANN approach to parameter determination is illustrated in Figure 2. In this case, the ANN operates as a black-box model of system Θ .

In this approach the following steps can be identified:

1. Output values F are obtained experimentally (see blocks 1–3).
2. An adequate model Ξ of system Θ , which performs operation (2), is created (block 5).
3. A representative set of points $\{P\}$ in the parametric space is generated (block 10) and the corresponding simulation values are calculated $\{F^*\}$. This set forms the training set $\{P, F^*\}$. (block 11).
4. The ANN is trained (block 12).
5. An initial estimation P_X^* is made for P_X (block 4).
6. An optimization algorithm, using a variation of parameters P_X^* , minimizes the discrepancy function $\|F^{**} - F\|$ (blocks 6–8 and 13).
7. Finally, the fitted parameters $P_X^* \approx P_X$ are obtained (block 9).

3. COMPUTATIONAL METHODS

3.1. Optimal Selection of Parameters for the Training Set. Being replaced by an approximating ANN, the simulation model is used only for initial training of the ANN. The step of the generation of the ANN training set now becomes the most time-consuming part in the proposed scheme in Figure 2, because to obtain each element of this set the relatively slow simulation should be executed. Furthermore it is of crucial importance for a good approximation to have a representative training set. Therefore to increase the efficiency of a training set, it is necessary to use an algorithm to generate a representative set of parameter points (each point corresponds to a single vector of parameters P describing the system), which are maximally spread in the multidimensional parameter space together with a minimal number of points. Furthermore, to be most flexible, the algorithm should make it possible to increase the number of points without any penalties.

In the present work, the following scheme was developed and applied for the selection of points. It is assumed that each parameter is normalized to the range $[0, 1]$.

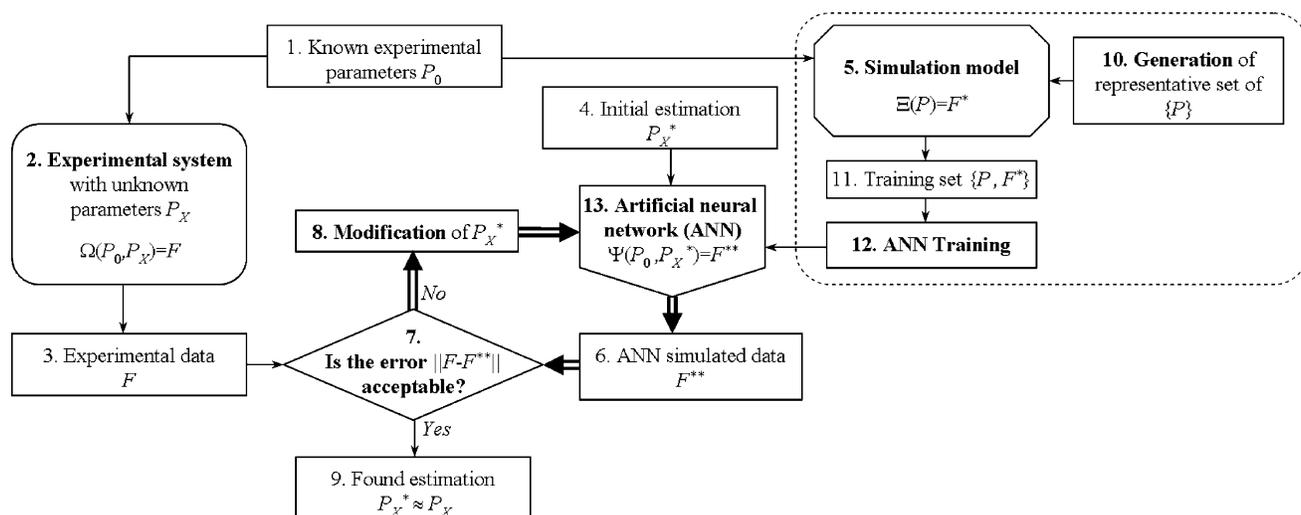


Figure 2. Flow diagram of simulation-based fitting with artificial neural network approximation. The dotted box shows the extension of the simulation-based fitting method (see Figure 1) by the artificial neural network fitting procedure.

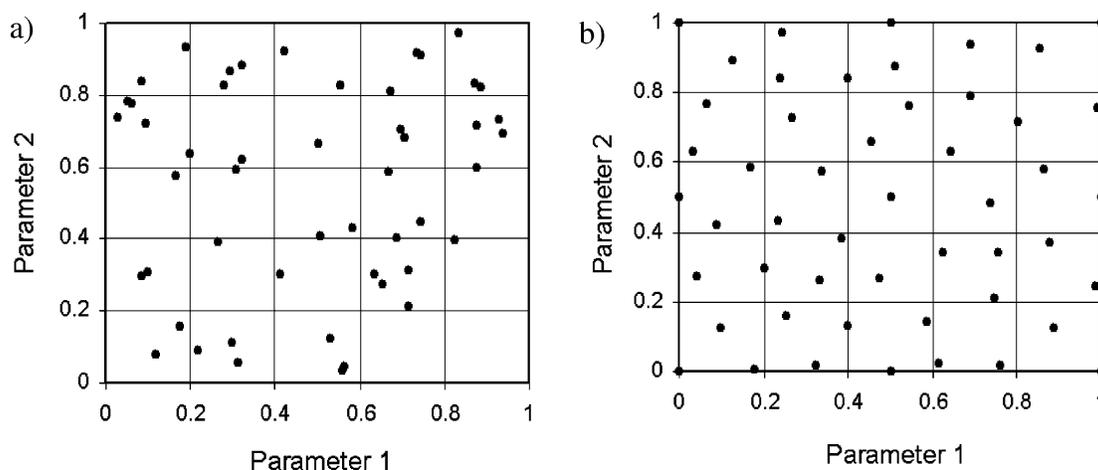


Figure 3. Illustration of the principle of point selection in a two-dimensional parameter space. Uniformly distributed random points are shown in a. Points obtained by the proposed algorithm are presented in b.

1. A set of “boundary” points was generated. For every parameter three values were taken: minimal, maximal, and mean. Then all their combinations were taken into account.

2. Main training set generation. Here points are chosen by the following algorithm:

2.1. Let n be the dimensionality of the parametric space and N the number of found points. The constant $a = 1$ is preset.

2.2. A point with random parametric coordinates is taken.

2.3. The distance d from the new point to all previously generated points is calculated.

2.4. If the following condition is true

$$\min(d) > \frac{a}{\sqrt[n]{N+1}} \quad (4)$$

the point is accepted and N is increased by 1. Else, the algorithm checks how many unsuccessful attempts were made before, and if there was a sufficient number of such attempts (in our experiments –1000), the value of a is decreased by 10%.

2.5. The stopping criterion is checked. If it is false, the algorithm goes to step 2.2.

To illustrate the scheme, a space of two parameters was taken ($n = 2$). The resulting points in comparison with randomly selected ones are shown in Figure 3. Obviously, the application of the scheme allows a uniform infill of the two-dimensional parameter space. The infill itself remains random and can easily be continued. Furthermore, the application of such an approach to the generation of points gives the possibility to select the most distant points of the control set during ANN training (see section 3.3) and allows one to avoid the selection of points with equal “parametric coordinates”. This is important for generalizing the ANN.

3.2. ANN Structure. In our research the optimal number of neurons was estimated using the exhaustive search method. In this method, the number of neurons in the first and second hidden layers were optimized. For each number of neurons, the ANN was trained for a fixed number of iterations and the resulting training error was calculated. To obtain a statistically valid value for the training error, the training was repeated independently for several times. After all

possible combinations of neuron numbers within the region of search, the one with the lowest training error was taken as optimal. The optimal number of neurons found depends on the complexity of the model. These numbers are given in Table 2 (see section 5).

3.3. Training of the ANN. Upon training of the multilayer perceptron two rather contradictory conditions should be satisfied. From the one side the mismatch between desired and obtained outputs should be decreased. From the other side an ANN should not lose its generalization abilities.¹³ An excessively long training results in a sliding of the ANN coefficients to a local minimum. This makes the approximation worse in points that do not belong to a training set. This phenomenon is called overtraining.^{12,13} Therefore, a special training strategy, based on the generation of an additional small control set, was applied to deal with this problem. After each epoch of training the performance of the ANN is verified on this control set. The training is terminated if the performance does not change or decreases for a certain number of epochs.¹² Numerical calculations were performed to determine the optimal size of the control set. The training set was separated into an actual training set and a control set with the following ratios: 90–10%, 85–15%, 80–20%, 75–25%, and 70–30%. For the three-parametric model the best performance was obtained for the 80–20% ratio, for the four-parametric model this was 75–15%, and for the five- and six-parametric model the ratio 90–10% gave the best results. The application of the proposed method of training set generation (section 3.1) allows a simple separation procedure: the last generated elements of the training set should be taken for the control set.

In our calculations the ANN is trained by the back-propagation error algorithm with the Levenberg–Marquardt optimization technique.¹⁴

4. EXPERIMENTAL OBJECTS AND METHODS

The proposed approach of the neural network approximation, as shown in Figure 2, was tested on a simulation model of fluorescence resonance energy transfer (FRET) between fluorescent labels of bacteriophage M13 major coat protein mutants incorporated into a lipid bilayer.

4.1. FRET. The idea of FRET spectroscopy is based on a dipole–dipole radiationless energy transfer and was initially developed by Förster¹⁵ and further enhanced by Stryer.¹⁶ Macromolecules studied (in our case, membrane proteins) are labeled with fluorescent probes of two types: donors and acceptors.⁹ The emission spectrum of the donor and the absorption spectrum of the acceptor should overlap. Donors are excited by an external light source, and some of them transfer excitation energy to acceptors due to dipole–dipole radiationless energy transfer. The probability of energy transfer for an isolated donor–acceptor pair is

$$p_{\text{ET}} = \frac{1}{1 + (r/R_0)^6} \quad (5)$$

where r is the distance between the donor and the acceptor and R_0 is the so-called Förster distance, which corresponds to 50% energy-transfer probability via dipole–dipole interaction.¹⁵ For a system containing N_a acceptors, the expression for the energy transfer becomes somewhat more complex:

$$p_{\text{ET}} = \frac{\sum_{i=1}^{N_a} (R_0/r_i)^6}{1 + \sum_{i=1}^{N_a} (R_0/r_i)^6} \quad (6)$$

The mean probability of energy transfer in the system, containing N_d donors and N_a acceptors, is called the energy-transfer efficiency and can be calculated as a mean value of energy-transfer probabilities for all donors:

$$E = \frac{1}{N_d} \sum_{j=1}^{N_d} (p_{\text{ET}})_j \quad (7)$$

By observing the energy-transfer process, one can get information about the relative location of donor and acceptor labels.

4.2. Biophysical Protein–Lipid Model. The membrane-bound major coat protein of M13 bacteriophage, which infects *E. coli*, is an excellent model system to study fundamental aspects of protein–lipid and protein–protein interactions.¹⁷ This single membrane-spanning protein consists of 50 amino acid residues and has mainly an α -helical conformation. The protein has been extensively studied in membrane model systems by biophysical techniques.^{17–20}

For FRET studies, the natural amino acid residue tryptophan of M13 major coat protein at position 26 was used as a donor label. To introduce an acceptor label to the protein, a number of mutants, containing unique cysteine residues at specific positions, was produced. The cysteine residues were specifically labeled with the fluorescent environmental probe *N*-(iodoacetylaminoethyl)-5-naphthylamine-1-sulfonic acid (AEDANS).²⁰ This fluorescent label was used as an acceptor. Since the labeling efficiency with AEDANS is less than 100%, the entire protein–lipid system contains proteins of two types: unlabeled proteins—with the natural donor—and labeled ones—with both donor and acceptor.

To study such a complex system the following simplified spatial model was designed. The biological membrane is approximated by a two-dimensional periodic structure with hexagonal packing of the lipids in which the M13 coat protein mutants are distributed (Figure 4a). The area occupied by each membrane protein on the membrane surface is assumed to be equal to that of two lipids. It is assumed that the distance between two nearest molecules on the grid is 8.0 Å and the thickness of the lipid bilayer is 30 Å. The α -helical M13 coat protein mutants are approximated by rods with a constant location of the donor (D) and a variable location of the acceptor (A) (see Figure 4b).

4.3. Simulation of Energy Transfer. The input parameters of the model are presented in Table 1. The ranges listed in this table are selected such that they are physically valid and cover all possible experimental situations.

In the experimental situation, parameters 1–3 and 6 are known (some of them within a small experimental inaccuracy); thus, they can be regarded as P_0 in terms of the parameter description in section 2. However, in general the separation of the parameters in Table 1 into P_0 and P_X depends on the situation. For instance, for the three-parametric model parameters 1 and 2 can be used as P_X ,

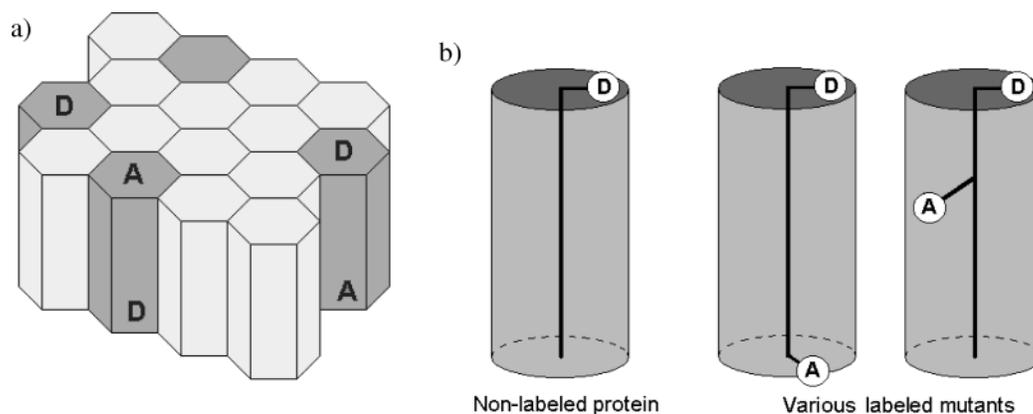


Figure 4. Model of a membrane (a) and a membrane protein (b) with fluorescent labels. The nonlabeled protein only contains a tryptophan. The protein mutants have an acceptor label (AEDANS) at various positions along the protein structure. The protein is assumed to be α -helical.

Table 1. Input Parameters of the Simulation Model

no.	parameter	description	range
1	surface density of labeled proteins	ratio of the area occupied by labeled proteins (containing the donor and acceptor) to the area of the entire membrane	0.0001 \leq 0.1
2	surface density of nonlabeled proteins	ratio of the area occupied by nonlabeled proteins (containing only donor) to the area of the entire membrane	0.0001 \leq 0.1
3	labeling site	amino acid residue number to which the acceptor is attached	1 \leq 50
4	coefficient of protein association	probability that a selected protein is located in the immediate proximity to another one	0 \leq 1
5	size of molecules	minimal distance between the centers of 2 nearest molecules (proteins and lipids)	5 \leq 10 \times 81
6	Förster distance	donor–acceptor distance (for an isolated pair) corresponding to 50% energy transfer	1 \leq 100 \times 81

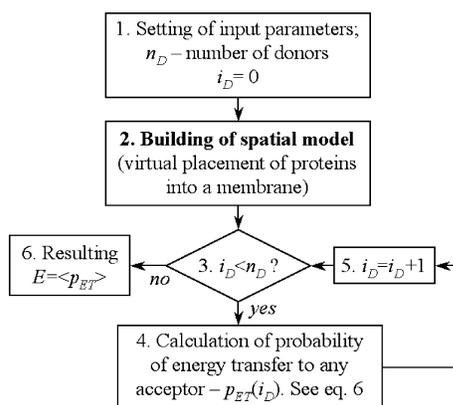


Figure 5. Flow diagram of a single simulation of energy transfer in the protein–lipid system.

and parameter 3, which is known precisely, as P_0 . It should be mentioned that the three-parametric model was used only for the validation of the methodology. For the four-parametric model, the coefficient of protein association becomes the subject of interest (P_X), while parameters 1–3 are the known parameters (P_0).

The fluorescence intensity and energy-transfer efficiency for the entire protein–lipid system are taken as output values (F in terms of the description in section 2). Because of the simulation nature of the model, the resulting output contains stochastic errors. Therefore simulations are run several times to reduce these errors. The flow diagram of the simulation is shown in Figure 5.

The simulation is carried out in the following way:

1. The parameters of the system are set (block 1).
2. A spatial model of the membrane with embedded proteins is created in accordance with the input parameters.

The coordinates and orientation of the proteins provide information about the locations of donors and acceptors in the system (block 2).

3. For each donor (denoted as i_D), the distances to all acceptor, are considered and the probability of energy transfer (to any of them) is calculated using eq 6 (blocks 3–5).

4. The mean probability of energy transfer among all donors results in the energy-transfer efficiency for the whole system.

5. Steps 2–4 (and blocks 2–6 in the flow diagram) are repeated for several times to decrease the effect of the randomness of the protein distribution. In our calculations we used an empirical value of $10^4/n_D$ simulations, where n_D is the number of donors.

Additional simulations using this model and experimental FRET data will be published elsewhere (Nazarov et. al., manuscript in preparation).

5. RESULTS AND DISCUSSION

5.1. ANN Configuration. Before transferring the simulation model to the ANN, the input parameters (Table 1) were normalized to the range [0,1] by the simple linear minimax method. Thus, input values 0 correspond to minimal possible parameter values and 1 to the maximal ones.

The optimal number of neurons obtained experimentally varied with the number of input parameters. These numbers are presented in Table 2.

After the number of neurons in the ANN was determined, it was trained as was described in section 3. To avoid overtraining, after each 10 epochs the ANN was tested on a control set. If the result of testing did not improve for 30 epochs, the training procedure was stopped. The resulting mean relative square error on the training set varied up to 2%.

Table 2. Optimal Number of Neurons in the ANN Used^a

no. of input values	3	4	5	6
model params	1-3	1-4	1-5	1-6
no. of neurons in the first layer	13	15	18	20
no. of neurons in the second layer	10	13	16	20

^a The input parameters are described in Table 1.

Table 3. Time Costs of the ANN Approximation of the FRET Simulation Model

no. of params	3	4	5	6
time for generation of the training set (h)	11	22	56	110
time for training (min)	6	10	14	20
time for ANN simulation (s)	6.0×10^{-4}	7.0×10^{-4}	8.0×10^{-4}	10^{-3}
av time for simulation modeling (s)	40	40	40	40
av gain in computer time	6.7×10^4	5.7×10^4	5.0×10^4	4.0×10^4

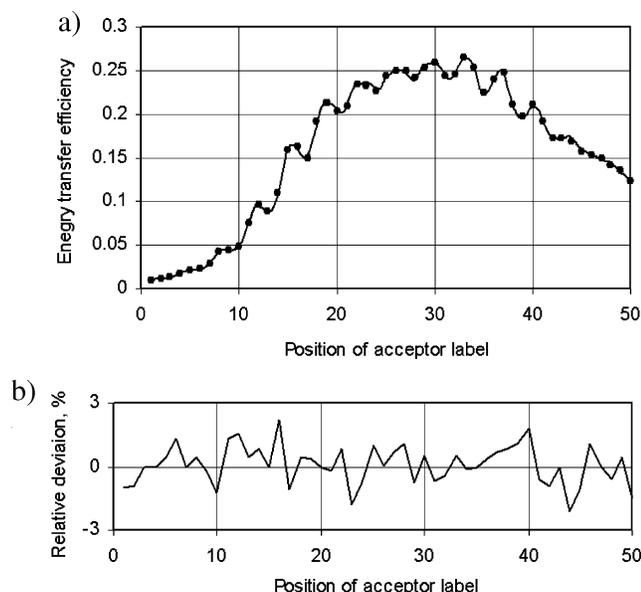


Figure 6. ANN approximation of the simulation model (a) and relative deviations of the ANN result and the simulation model (b). In a, the circles show the result of the simulation modeling and the line is the ANN approximation.

5.2. Time Costs. All calculations in this article were made in MATLAB 6.1 with the Neural Networks Toolbox on a PC with Intel Pentium III-850 CPU. The time costs of the ANN method application in this set up are shown in Table 3.

From Table 3, it is clear that the generation of training sets is the most time-consuming operation. However, it should be noted that this process does not need the supervision of a human and the training itself needs to be performed only once for each simulation model. The gain in computing time, which is about 5×10^4 , does not decrease significantly with the increase of the network complexity.

A typical illustration of the ANN approximation of the simulation model is shown in Figure 6. Here the energy-transfer efficiency is plotted as a function of the location of the acceptor. The oscillations in this plot arise from the α -helical nature of the protein model, as is shown in Figure 4b. The relative deviation between the ANN approximation and the actual model calculation is less than 3%, showing that the ANN approximation performs very well.

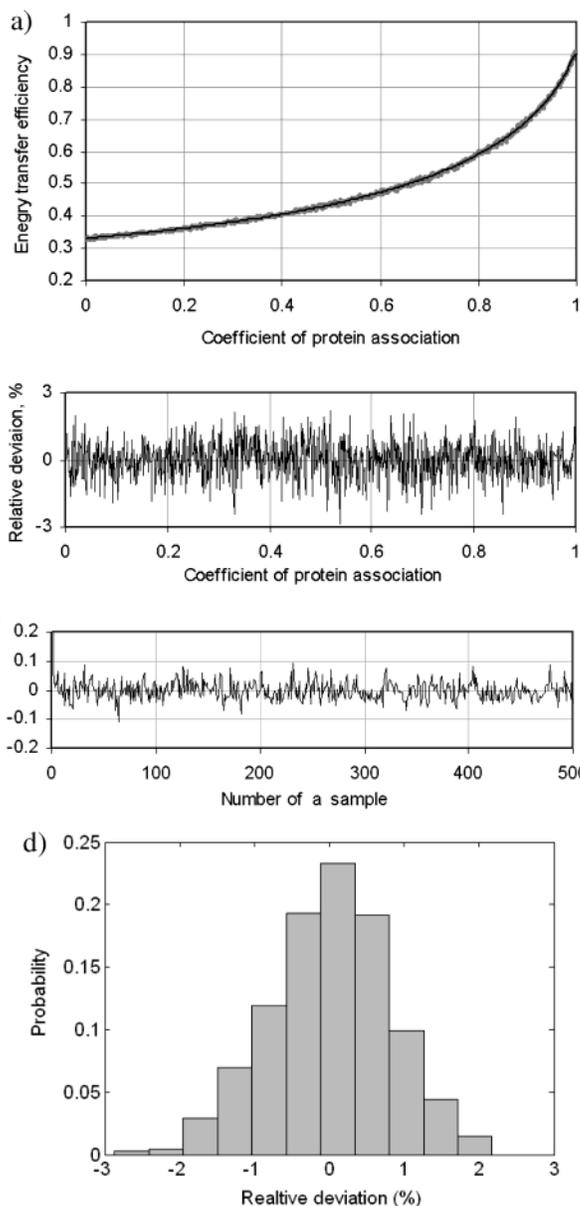


Figure 7. Consistency of the ANN approximation. In a, the thick gray line is the result of the simulation modeling and the thin black line is the ANN approximation. Below this graph the relative deviations between simulation model and approximation (b) and their autocorrelation function (c) are given. The distribution of deviations is given in d.

5.3. Consistency of the Approximation. To obtain information about the consistency of the ANN approximation, we conducted several statistical calculations on the four-parametrical model. In these calculations we modified one of the model parameters—the association coefficient—and analyzed the deviation between the ANN approximation and the simulation modeling. In Figure 7a the energy-transfer efficiency is plotted for various values of the protein association constant. The agreement between the ANN approximation and actual model calculation is good.

Since the autocorrelation function, calculated from the deviations (Figure 7a), is close to a δ function, we conclude that the deviations are not correlated and behave stochastically. The distribution of the deviations (Figure 7d) is close to a Gaussian line shape, indicating that the deviations are the result of the randomness of the simulation model.

It should be mentioned that a multilayer perceptron with sigmoid activating functions produces a smooth approximation of a stochastic simulation model. This approximation does not contain stochastic noise. Thus, the fitting procedure operates with a less stochastic discrepancy function $\|F - F^*\|$ and therefore contains less local minima than in the case of stochastic simulation model fitting.

6. CONCLUSIONS

The use of a trained ANN in the biophysical modeling presented here results in a gain in computing time by a factor of 5×10^4 . Moreover an ANN produces a smooth approximation of the results of a stochastic simulation. Thus, it decreases the level of stochastic errors. Due to this smooth dependency it will simplify the application of standard optimization techniques, such as gradient search, for parameter determination. It was shown that the deviations between the actual model outputs and its ANN approximation have a stochastic nature. In our case the relative deviation was less than 3%.

The approach used in our calculations has some imperfections. It works only when the number of variable parameters is relatively small (in our calculations up to 6). Furthermore the calculations related to the generation of the training set are quite time extensive, although they need to be performed only once for a simulation model.

In conclusion, the method of ANN modeling is an excellent tool for determination of parameters of specific systems. In fact the method can be generalized to analyze any experimental system for which SBF can be applied.

REFERENCES AND NOTES

- (1) Mendes, P.; Kell D. B. Nonlinear optimization of biochemical pathways: Application to metabolic engineering and parameter estimation. *Bioinformatics* **1998**, *14*, 869–883.
- (2) Yatskou, M. M.; Donker, H.; Novikov, E. G.; Koehorst, R. B. M.; van Hoek, A.; Apanasovich, V. V.; Schaafsma, T. J. Nonisotropic excitation energy transport in organized molecular systems: Monte Carlo simulation-based analysis of time-resolved fluorescence. *J. Phys. Chem. A* **2001**, *105*, 9498–9508.
- (3) Yatskou, M. M.; Donker, H.; Koehorst, R. B. M.; van Hoek, A.; Schaafsma, T. J. Energy transfer processes in zinc porphyrin films studied by time-resolved fluorescence spectroscopy and Monte Carlo simulations. *Chem. Phys. Lett.* **2001**, *345*, 141–150.
- (4) Berney, C.; Danuser, G. FRET or no FRET: A quantitative comparison. *Biophys. J.* **2003**, *84*, 3992–4010.
- (5) Frederix, P.; de Beer, E. L.; Hamelink, W.; Gerritsen H. C. Dynamic Monte Carlo simulations to model FRET and photobleaching in systems with multiple donor–acceptor interactions. *J. Phys. Chem. B* **2002**, *106*, 6793–6801.
- (6) Apanasovich, V. V.; Novikov, E. G.; Yatskov, N. N. Analysis of fluorescence decay kinetics of complex molecular systems using the Monte Carlo method. *J. Appl. Spectrosc.* **2000**, *67*, 612–618.
- (7) Bishop, C. M. *Neural networks for pattern recognition*; Oxford University Press: Oxford, U.K., 1995.
- (8) Wasserman, P. D. *Neural computing theory and practice*; Van Nostrand Reinhold: New York, 1989.
- (9) Lakowicz, J. R. *Principles of fluorescence spectroscopy*; Kluwer Academic/Plenum: New York, 1999.
- (10) Cybenko, G. Approximations by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **1989**, *2*, 304–314.
- (11) Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Networks* **1989**, *2*, 359–366.
- (12) Tetko, I. V.; Livingstone, D. J.; Luik, A. I. Neural network studies. 1. Comparison of overfitting and overtraining. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 826–833.
- (13) Stegemann, J. A.; Buenfeld, N. R. A glossary of basic neural network terminology for regression problems. *Neural Comput. Appl.* **1999**, *8*, 290–296.
- (14) Hagan, M. T.; Menhaj, M. B. Training feedforward networks with the Marquardt algorithm. *IEEE Trans. Neural Networks* **1994**, *5*, 989–993.
- (15) Förster, T. Intermolecular energy migration and fluorescence. *Ann. Phys.* **1948**, *2*, 55–75.
- (16) Stryer, L. Fluorescence energy transfer as a spectroscopic ruler. *Annu. Rev. Biochem.* **1978**, *47*, 819–846.
- (17) Meijer, A. B.; Spruijt, R. B.; Wolfs, C. J. A. M.; Hemminga, M. A. Membrane-anchoring interactions of M13 major coat protein. *Biochemistry* **2001**, *40*, 8815–8820.
- (18) Stopar, D.; Spruijt, R. B.; Wolfs, C. J. A. M.; Hemminga, M. A. Structural characterization of bacteriophage M13 solubilization by amphiphiles. *Biochim. Biophys. Acta* **2002**, *1594*, 54–63.
- (19) Stopar, D.; Spruijt, R. B.; Wolfs, C. J. A. M.; Hemminga, M. A. Protein–lipid interactions of bacteriophage M13 major coat protein. *Biochim. Biophys. Acta* **2003**, *78451*, 1–11.
- (20) Spruijt, R. B.; Meijer, A. B.; Wolfs, C. J. A. M.; Hemminga, M. A. Localization and rearrangement modulation of the N-terminal arm of the membrane-bound major coat protein of bacteriophage M13. *Biochim. Biophys. Acta* **2000**, *1509*, 311–323.

CI034149G