

Application of Neural Networks to Population Pharmacokinetic Data Analysis

HSIAO-HUI CHOW^{†x}, KRISTIN M. TOLLE[‡], DENISE J. ROE[§], VICTOR ELSBERRY[†], AND HSINCHUN CHEN[‡]

Received September 24, 1996, from the [†]Department of Pharmacy Practice and Science, [‡]Department of Management Information Systems, and [§]Arizona Cancer Center, University of Arizona, Tucson, AZ 85721. Accepted for publication March 24, 1997[®].

Abstract □ This research examined the applicability of using a neural network approach to analyze population pharmacokinetic data. Such data were collected retrospectively from pediatric patients who had received tobramycin for the treatment of bacterial infection. The information collected included patient-related demographic variables (age, weight, gender, and other underlying illness), the individual's dosing regimens (dose and dosing interval), time of blood drawn, and the resulting tobramycin concentration. Neural networks were trained with this information to capture the relationships between the plasma tobramycin levels and the following factors: patient-related demographic factors, dosing regimens, and time of blood drawn. The data were also analyzed using a standard population pharmacokinetic modeling program, NONMEM. The observed vs predicted concentration relationships obtained from the neural network approach were similar to those from NONMEM. The residuals of the predictions from neural network analyses showed a positive correlation with that from NONMEM. Average absolute errors were 33.9 and 37.3% for neural networks and 39.9% for NONMEM. Average prediction errors were found to be 2.59 and -5.01% for neural networks and 17.7% for NONMEM. We concluded that neural networks were capable of capturing the relationships between plasma drug levels and patient-related prognostic factors from routinely collected sparse within-patient pharmacokinetic data. Neural networks can therefore be considered to have potential to become a useful analytical tool for population pharmacokinetic data analysis.

Introduction

In clinical practice, collection of sparse within-patient concentration-time data along with the clinically relevant prognostic factors and the dosing history are routinely conducted as part of therapeutic monitoring for a variety of drugs. Population pharmacokinetic data analysis techniques have been developed to allow analysis of such data to gain further insight into population pharmacokinetic characteristics and the effects of different demographic factors on the behavior of a drug. It has been realized that, since only a few concentration measurements need to be collected from each patient for analysis, collection of this type of data could be easily incorporated into all phases of the drug development process to maximize the amount of knowledge that could be discovered.

In recent years, a number of population modeling programs have become available.¹ Among them, the NONMEM program was the first developed and has been used most extensively to analyze actual clinical pharmacokinetic data. Building a population pharmacokinetic model requires understanding and selection of various mathematical/statistical models that include a pharmacokinetic structure model relating dose, sampling time, and pharmacokinetic parameters to plasma drug levels, regression models for relationships between patient characteristics and the pharmacokinetic parameters, a population model for intersubject variability, and

a variance model for random residual variation in the data (inpatient variability).^{2,3} This can be a tedious and time-consuming process.

Neural networks are computational systems which are developed to simulate the neurological processing abilities of biological systems and are known for their adaptive learning and self-organizational capabilities. Application of this technique to biopharmaceutical data analysis has received considerable attention recently.⁴⁻⁷ Brier et al.⁸ have examined the use of neural networks for population pharmacokinetic data analysis and concluded that neural networks and NONMEM provided comparable predictions of plasma drug concentrations. The strength of neural networks is that they do not assume a specific model. Instead, they learn to establish the input and output relationships from the data provided to them. This greatly simplifies the modeling work involved in traditional population pharmacokinetic data analysis. In this study, we further explored the applicability of the neural network approach to population pharmacokinetic data analysis to capture the relationship between patient-related prognostic factors and plasma drug levels, using data compiled from our clinical practice site. A comparison was made between the results from the neural network approach and those obtained using the NONMEM program.

Methods

Data Compilation—The data were obtained retrospectively from the clinical dosing services of the Department of Pediatrics of the University Medical Center in Tucson, AZ. Patient-related information was collected from the charts of 101 pediatric patients who, between 1983 and 1992, received multiple doses of tobramycin infusion for the treatment of bacterial infection. This information included age, weight, gender, other underlying illness (cystic fibrosis or cancer), dose, dosing interval, time of blood drawn, and the resulting serum tobramycin concentration. Blood samples were collected at two time points (roughly at the peak and trough) after the fourth or the fifth dose of infusion. Steady-state should have been attained before the collection of the blood samples.

Some of the patients had received more than one dosing regimen of tobramycin treatment during one hospital stay and a sizable number of patients had received tobramycin treatment during separate infection episodes. This resulted in the availability of multiple measurements of tobramycin levels in these patients. There was, therefore, a total of 311 pairs of peak and trough tobramycin concentration data which provided 622 tobramycin concentration measurements.

NONMEM—The entire data collection was analyzed using NONMEM version IV, level 2.1. Since steady-state levels should have been attained after the fourth or the fifth dose, a one-compartment model with intravenous infusion and first-order elimination was used. Various regression models relating different combinations of clinical factors (i.e., age, weight, gender, and illness) to pharmacokinetic parameters were examined, and the model that provided the smallest objective function was selected. Various statistical models describing the between-patient and within-patient variability were also tested. The final models used in the NONMEM program are shown below:

[®] Abstract published in *Advance ACS Abstracts*, May 15, 1997.

the pharmacokinetic structure model

$$\hat{C}_{ij} = \frac{K_{oi}(1 - e^{-(CL_i/Vd_i)T_i})}{CL_i(1 - e^{-(CL_i/Vd_i)\tau_i})} e^{-(CL_i/Vd_i)t'_{ij}}$$

where for child i , \hat{C}_{ij} = predicted concentration at measurement j , K_{oi} = infusion rate, T_i = infusion time (0.5 h), τ_i = dosing interval, t'_{ij} = elapsed time at measurement j after end of infusion, CL_i = clearance, and Vd_i = volume of distribution.

the regression model

$$TVCL = \theta_1 \ln(\text{weight}) + \theta_3 \times \text{age}$$

$$TVVd = \theta_2 \ln(\text{weight})$$

where TVCL is the typical value of clearance; TVVd is the typical value of volume of distribution; and θ s are model parameters.

the population model

$$CL_i = TVCL \times e^{\eta_1}$$

$$Vd_i = TVVd \times e^{\eta_2}$$

where η s are the interindividual random effect variables and the variances of η s are model parameters.

the variance model

$$\ln C_{ij} = \ln \hat{C}_{ij} + \epsilon_{ij}$$

where C is the observed concentration and \hat{C} is the concentration predicted from the pharmacokinetic structure model, ϵ is the intraindividual random effect variable, and the variance of ϵ is a model parameter.

Because of the nature of the program, different time events including time of initiation of the treatment, time of initiation of each infusion, time of the end of each infusion, and time of blood drawn were provided to the program. These times were converted to accumulated values counting from the initiation of the tobramycin treatment for any given hospital stay. The tobramycin concentrations were converted to logarithmic values. The data derived within one hospitalization for any given patient were entered into the program as a set of data. After the final models were selected and converged, predicted tobramycin concentrations were obtained from NONMEM.

Neural Networks—A backpropagation algorithm implemented in C language (running on a Pentium-based PC, 133 MHz) was developed. The specific algorithm adopted in this research has been described previously.^{6,9} Figure 1 illustrates a schematic representation of the neural network used to predict drug concentration from information on clinical prognostic factors, dosing regimens, and elapsed time after dosing. This network had seven input variables (i.e., age, gender, illness, weight, dose, dosing interval, and time of blood drawn). The time of blood drawn was presented in two different fashions, one as accumulated time counting from the initiation of the tobramycin treatment (test I) and the other as time after the initiation of the fourth or the fifth infusion within one dosing interval (test II). The values of age, weight, dose, dosing interval, or time of blood drawn were normalized linearly to range between 0 and 1. Gender and illness variables had values of 0 or 1, with male = 0, female = 1 and oncology = 0, cystic fibrosis = 1, respectively. The logarithm of the tobramycin concentration was used as output values, which were normalized according to the following formula to result in output values ranging from 0 to 1.

normalized output value =

$$\frac{[\ln(\text{observed concn}) - \ln(\text{smallest observed concn})]}{[\ln(\text{largest observed concn}) - \ln(\text{smallest observed concn})]}$$

Flexible selection of network parameters such as the number of hidden units, learning rate, and momentum factor was allowed. After experimentation, eight and seven hidden nodes were selected to

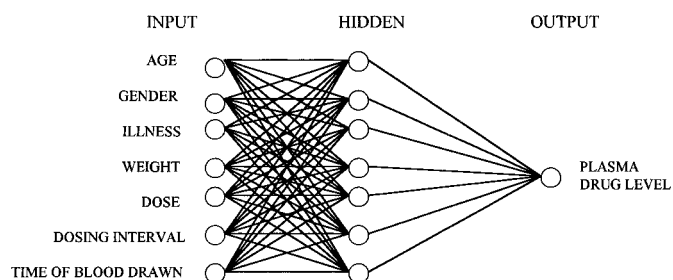


Figure 1—Schematic representation of the neural network used in estimating plasma tobramycin levels from demographic factors and dosing regimens.

provide the interconnections between input and output for neural networks test I and test II, respectively. As shown in Figure 1, each node in a layer is connected to every node in the next layer. Each interconnection between nodes is associated with a weighting factor.

Each of the 622 tobramycin measurements along with its corresponding clinical information, dosing regimens, and time of blood drawn was treated as a new input and output data pair. Data associated with the same patient were arranged in adjacent order when they were used to train the network. Of the 622 sets of input and output data, the first two-thirds of the data (415 sets of data) were from patients who received their first tobramycin treatment before 1989 and were used as a training data set. The last one-third of the data (207 sets of data) were from patients who received their first tobramycin treatment after 1989 and were used as the tuning set. Because of subsequent hospitalizations, some of the patients in the training data set had had tobramycin plasma levels collected after 1989. Furthermore, no apparent time dependent differences in the patient population could be observed to have resulted in significant differences in these two data sets.

The networks were trained with the input–output (prognostic factors and dosing regimens vs tobramycin concentrations) data pairs in the training set. The network started out with a random set of connection weights. Each pair of data went through two stages of activation: a forward pass and a backward pass. The forward pass involved presenting a sample input to the network and letting activation flow until the output layer was reached. During the backward pass, the network's actual output from the forward pass (predicted concentrations) was compared with the target output (observed concentrations), and errors were computed for the output units. The weights connected to the output units were adjusted in order to reduce those errors. The error estimates of the output units were then used to derive error estimates for the units in the hidden layers. Finally, errors were propagated back to the connections stemming from the input units. The delta rule⁹ was used as the error correction formula to adjust the connection weights each time the network saw a new input–output pair. A complete round of forward–backward passes and weight adjustments using all input–output pairs in the training data set is called an epoch. A backpropagation network needs to “learn” from the same data set through a few hundreds—sometimes even thousands—of epochs in order to gradually refine its connection weights.

In this work, after each epoch of training and adjustment of the connection weights, the trained network was evaluated with the tuning data. The errors of the output values from the tuning data were calculated and reported. These errors were not propagated back for connection weight adjustments. Convergence of training (termination of training) was determined when the network showed a minimum sum of square error in predicting the tobramycin concentrations in the tuning data set.

Statistical Analysis—The observed and predicted logarithm of tobramycin concentrations were transformed back to actual concentrations and then used to evaluate and compare the performances of the neural network tests and of NONMEM. Mean squared error (MSE), mean prediction error (ME), and their corresponding 95% confidence intervals were estimated according to that described by Sheiner and Beal.¹⁰ MSE and ME were used as measures of precision and bias, respectively. Precision and bias were also evaluated relative to the observed tobramycin concentrations using percent absolute error (%AE) and percent prediction error (%PE), respectively. These percent errors were estimated according to the following equa-

Table 1—Demographic Information for the Pediatric Patients from Whom the Data Were Compiled

	Cystic Fibrosis	Oncology	Total
Number of children	46	55	101
Gender			
Male	23	28	51
Female	23	27	50
Age (years)			
Mean	9.0	7.7	8.6
Standard Deviation	5.1	5.0	5.1
Median	9.3	7.0	8.4
Weight (kg)			
Mean	24.7	26.8	25.3
Standard Deviation	12.7	19.0	15.0
Median	22.9	20.0	22.0
Number of concentration pairs	213	98	311

tions:

$$\%AE = \frac{|\text{predicted concn} - \text{observed concn}|}{\text{observed concn}} \times 100 \quad (1)$$

$$\%PE = \frac{(\text{predicted concn} - \text{observed concn})}{\text{observed concn}} \times 100 \quad (2)$$

Statistical measures of the performances of the neural network tests and NONMEM were compared using a paired *t*-test. A *p* value <0.05 was considered to be statistically significant.

Results

The descriptive statistics of the demographic characteristics of the data are presented in Table 1. There were similar numbers of cystic fibrosis patients and oncology patients in the database. Around 50% of each patient population was female. The average age of cystic fibrosis patients was older than that of oncology patients. Comparable average weights were observed for both populations. Because cystic fibrosis patients were often treated with tobramycin for multiple episodes of infection, twice as many tobramycin concentration data pairs were available from this population than from oncology patients.

In neural network analysis, experimentation was performed to define three different parameters associated with the training of a backpropagation neural network. These were learning rate, hidden units, and momentum. We performed the analysis by varying the values of these parameters, with the final setting selected as the set that provided the lowest sum of square error. The optimal learning rate, hidden units, and momentum selected for neural network tests I and II were 0.2, 8, 0 and 0.25, 7, 0, respectively. As described in the Method section, the training was terminated when the minimum sum of square error on the tuning data was attained. The number of training epochs reflected when the training was terminated, and this parameter differed significantly between tests I and II (2160 and 5680 for tests I and II, respectively).

Figure 2 illustrates the relationship between the predicted and observed concentrations from NONMEM and neural network analyses. In NONMEM analysis, the predicted concentration correlated well with the observed concentrations. The regression line between the predicted and observed concentrations deviated slightly from the line of identity. Similar to the NONMEM analysis, the predicted concentration from both tests of neural network analyses correlated well with the observed concentrations. The regression line between the predicted and observed concentrations deviated

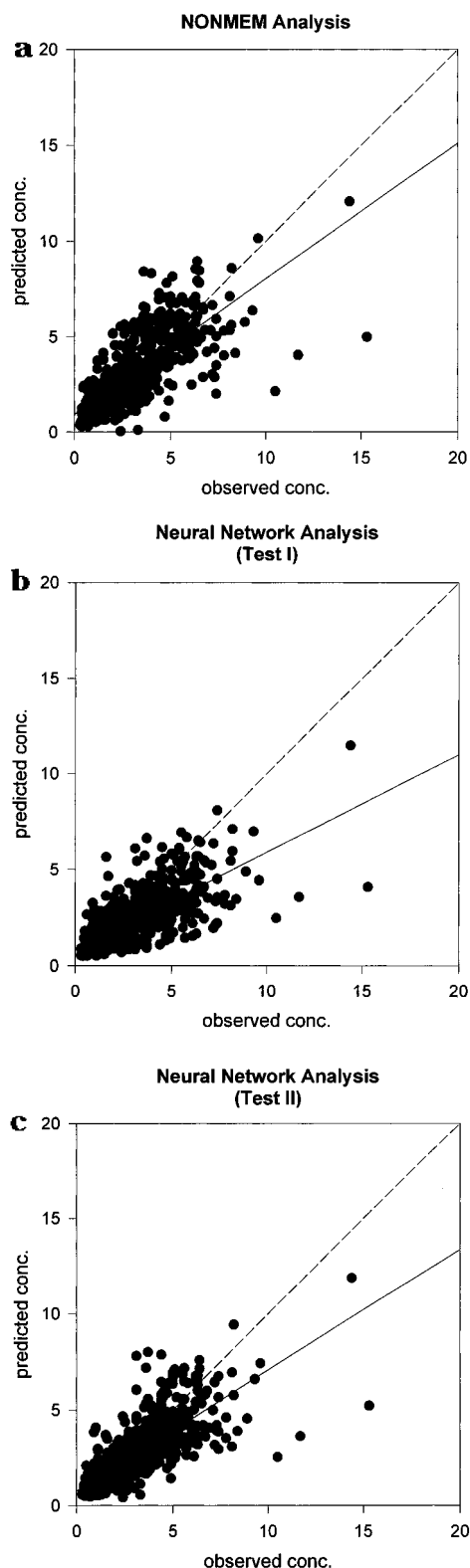


Figure 2—Plots of observed and predicted tobramycin levels from the neural network approach and the NONMEM program. Neural network test I used accumulated times within one hospitalization, whereas test II was trained with un-accumulated times within one dosing interval. Solid line represents the linear regression line of observed and predicted concentrations [(a) NONMEM, $y = 0.929 + 0.708x$, $r^2 = 0.58$; (b) neural network test I, $y = 0.818 + 0.509x$, $r^2 = 0.52$; neural network test II, $y = 0.762 + 0.631x$, $r^2 = 0.60$] and dashed line represents the line of identity. Concentrations are in micrograms/milliliter.

more from the line of identity in the neural network analyses than did that in NONMEM analysis.

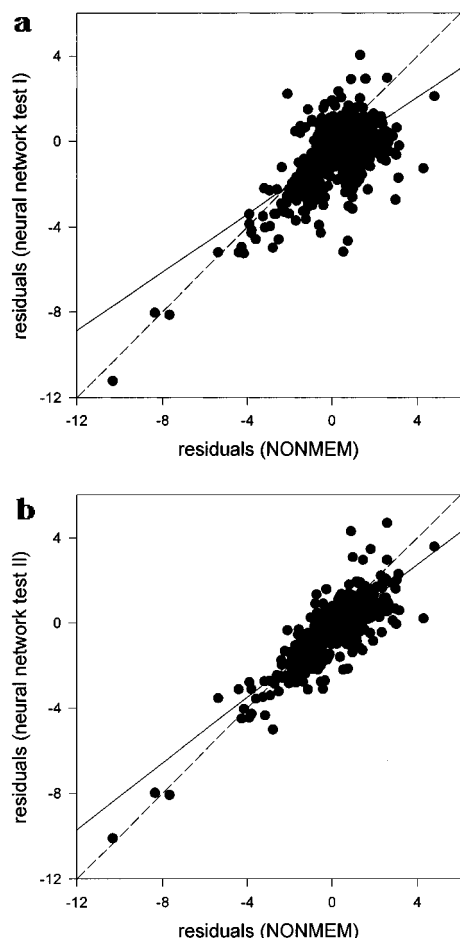


Figure 3—Plots of residuals of the prediction from the neural network approach versus that from the NONMEM program. Residuals were calculated as the difference between the predicted and observed concentrations. Neural network test I used accumulated times within one hospitalization, whereas test II was trained with unaccumulated times within one dosing interval. Solid line represents the linear regression line of residuals of the two methods [$y = -0.670 + 0.683x$, $r^2 = 0.44$ for the plot of neural network test I vs NONMEM (a) and $y = -0.378 + 0.776x$, $r^2 = 0.67$ for the plot of neural network test II vs NONMEM (b)] and dashed line represents the line of identity. Residual concentrations are in micrograms/milliliter.

Figure 3 demonstrates the relationship between the residuals of the predictions (estimated from predicted concentration – observed concentration) from NONMEM versus that from neural network analyses. Most of the residuals were found to range from -4 to 4 . It appears that the residuals of the predictions of the neural network tests showed a positive correlation with those of NONMEM, with a higher correlation found between neural network test II and NONMEM (correlation of coefficient, r , of 0.82).

Table 2 summarizes the statistical measures of the performance of neural network tests and NONMEM. When accumulated times were used, neural network test I resulted in significantly higher MSE than NONMEM analysis. Without the use of accumulated times, neural network test II provided precision of the predicted concentrations comparable to that with NONMEM analysis. When the precision was measured relative to the observed concentration (i.e., %AE), a better average precision was seen with neural network test II than with NONMEM (33.9% vs 39.9%). NONMEM results showed a smaller ME than that of the neural network analyses. However, when the bias was measured relative to the observed concentration (i.e., %PE), neural network test II exhibited only a 2.59% mean prediction error, whereas NONMEM had a 17.3% mean prediction error.

Table 2—Statistical Measures of the Performance of Neural Networks and the NONMEM Program

	NONMEM	Neural Network	
		Test I	Test II
Mean squared error (MSE)	1.88 (1.39, 2.37) ^a	2.38 ^b (1.81, 2.96)	1.78 ^c (1.30, 2.26)
Mean prediction error (ME)	0.077 (-0.031, 0.18)	-0.62 ^b (-0.73, 0.51)	-0.32 ^{b,c} (-0.42, 0.22)
% mean absolute error (mean AE)	39.9 (36.4, 43.3)	37.3 (34.6, 40.1)	33.9 ^{b,c} (31.2, 36.6)
% mean prediction error (mean PE)	17.3 (12.8, 21.7)	-5.01 ^b (-9.04, -0.98)	2.59 ^{b,c} (-1.23, 6.42)

^a Data in parentheses represent the 95% confidence interval of the mean. ^b Significantly different from NONMEM analysis, $p < 0.05$. ^c Significantly different from neural network test I, $p < 0.05$.

Discussion

This work showed that neural networks are capable of capturing the relationships between plasma drug levels and the following variables, patient-related prognostic factors, dosing regimens, and time of blood drawn from routinely collected sparse within-patient pharmacokinetic data. This research further documents the applicability of neural networks to population pharmacokinetic data analysis. It also showed that proper presentation of the time factor in the neural network approach contributed to improved concentration predictions.

Because pharmacokinetic processes are recognized to be highly time-dependent events, the analysis of plasma concentration data always requires consideration of the relative times of collection of blood samples and administration of the dose. Previous related work by Brier et al.⁸ could not consider this factor since the actual timing of the peak and trough samples had not been recorded. In the current research, the time of blood drawn was originally presented to the network as elapsed times accumulated from the initiation of the therapy within one hospitalization (test I). This was done to match the data input into the NONMEM program. Because of the nature of the NONMEM program, information on different time events including time of the initiation of the tobramycin treatment, time of the initiation or the termination of each infusion, and time of blood drawn was provided to the program. These times were converted to accumulated values counting from the initiation of the therapy. However, internally within the NONMEM program the concentration-time profile was analyzed within each dosing interval, i.e., the blood sampling times were converted back to the values within each dosing interval. In the neural network approach, when the accumulated time values were presented to the network, the differences between the times of the pair of blood samples collected within one dosing interval became very small when compared to the total accumulated times. The network therefore attempted to capture the global relationship over the total accumulated time, which resulted in less satisfactory performance of the estimation of tobramycin concentrations within a dosing interval. When the nonaccumulated times were used as the input data (test II), the performance of the network improved, resulting in smaller errors in the predicted concentrations than those from the NONMEM program and neural network test I. Addition of input variables which could account for the accumulated nature of the time factor (e.g., the total number of doses received) could improve the performance of neural network test I. Comparison of the two neural network tests suggests that the way that the time factor is presented to the network is important to the network's performance on time-dependent events.

In our neural network analysis, experimentation was performed to define three different parameters associated with the training of a backpropagation neural network: learning rate, hidden units, and momentum. Although the selection of these parameters could sometimes be empirical, it is recognized that each of these choices could have a significant effect on network performance. The number of hidden units refers to the number of nodes which are placed between the input and output layers to provide the interconnections between the input information and output values. A large number of hidden units can map a complex input-output relationship, but a network will consequently require more computation time and may suffer in its ability to generalize for an unknown data set. Learning rate and momentum, with values that could range from 0 to 1, are network parameters affecting the magnitude of weight adjustment between input, hidden, and output layers. The following simplified relationship presented by Erb¹¹ points out the effects of these two parameters on the weight adjustment:

$$\text{new weight change} = \text{learning rate} \times \text{error} + \text{momentum} \times \text{last weight change}$$

The higher the learning rate, the faster the learning process. However, if the learning rate is too high, oscillation in the weight change can prevent convergence to an optimal solution. Momentum is a parameter which takes the last weight change into consideration when making new weight changes. The higher the momentum value, the faster the neural network's training pace. But, similar to the impact from a high learning rate, high momentum results in fluctuation of new weight changes that can also impede convergence to an optimal solution. We have examined the performance of the network when the number of hidden units was increased from 1 to 60 for different learning rates (0.05–0.60) and momentum factors (0–0.90). As in our previous research,⁶ we found that a moderate learning rate without the use of a momentum factor provided the optimum network. Furthermore, minimal errors were obtained when the network contained seven to eight units in the hidden layer.

Over-training of neural networks has often created an undesirable effect on the generalization process. Upon repeated training, the system continues to minimize the differences between the predicted and observed concentrations of the training data. If training is continued, the mean square error may eventually be lower than that from NONMEM analysis. However, the network actually memorizes the input and output relationships of the data from the training set and eventually loses its capability to predict unknown data. We adopted a method suggested by Rumelhart¹² to incorporate a tuning (or validation) data set to prevent generalization problems. This technique also had been successfully adopted in our previous work to improve the predictive power of a trained network for unknown data.⁶ In the current application, this also provided a fair comparison with the results obtained from the NONMEM analysis. After each epoch of training, the trained network was evaluated with the tuning data. The sum of square error was estimated. The epoch that gave the lowest sum of square error for the tuning set was selected as the "optimum" epoch and the training was terminated. The final network with the optimized connection weights was used to estimate tobramycin concentrations. With this early termination of the training, the mean squared error and average percent of absolute error from neural network test II were comparable to or smaller than those from the NONMEM program.

Because tobramycin is extensively cleared by the kidney, it would be expected that creatinine clearance would serve as a major covariate in the population pharmacokinetic data

analysis. However, since serum creatinine levels had not been compiled from these patients, creatinine clearance could not be estimated and used as a clinical variable in our analyses. Previous work by Brier et al.⁸ included both serum creatinine and creatinine clearance as the input variables in the application of neural networks to prediction of gentamycin concentrations. Creatinine clearance also was used as the covariable for systemic clearance in their NONMEM analysis of the gentamycin data. It was found that NONMEM and neural networks exhibited average absolute errors of 18.6% and 16.5% in predicting the peak gentamycin concentrations of a set of leave-out data, respectively, and 59% and 48.3% in predicting the trough gentamycin concentrations, respectively. In our research, the performances of the two methods were not evaluated separately for the peak and trough concentrations because the blood samples were not drawn exactly at those time points. The average absolute error on the overall data was 39.9% from NONMEM and 33.9% from neural network test II, despite the lack of information on creatinine clearance. Furthermore, the residuals of the predictions from neural network test II were highly correlated with those from NONMEM. Consistency in residuals observed between different prediction methods tends to suggest that the residuals are likely to be attributed to errors in data recording or lack of certain common covariables. This observation suggests that not having included information on creatinine clearance in this comparative study should not have biased the performance of either the neural network analyses or the NONMEM analysis.

The major strength of the use of neural networks in the analysis of sparse within-patient pharmacokinetic data is that they do not assume a specific model. Instead, neural networks learn from the data to establish the input-output relationship. This eliminates the need to deal with the complex nonlinear mixed effect models required for the population pharmacokinetic modeling approach. With the incorporation of a hidden layer and fully connected units between layers, neural networks are capable of capturing complex nonlinear relationships and polynomial surfaces, including third or higher order terms as well as cross-product terms corresponding to interactions between clinical factors. From the standpoint of modeling and theoretical scientists, the major advantage of the neural networks could turn into its major weakness. Without a specific underlying model, neural networks will not be capable of providing population values of pharmacokinetic parameters and the within-patient and between-patient variability of these parameters. However, even in the traditional population pharmacokinetic modeling approach, insufficient information from the data may prevent accurate estimation of some of these parameters and questions regarding the models' reliability could still exist. Because the ultimate goals of population pharmacokinetic analysis during the process of drug development are to identify potential prognostic factors which could affect the resulting plasma drug levels and to recognize a special patient population which will handle the drug differently, incorporating other statistical analyses in determining the effects of demographic factors on the plasma drug levels will provide practical information needed during the process of drug development.

In conclusion, this work showed that neural networks were capable of capturing the relationships between patient-related factors and plasma drug levels from routinely collected sparse within-patient pharmacokinetic data. This further documents the applicability of neural networks to population pharmacokinetic data analysis. Our ongoing work involves incorporation of a graphical interface and what-if analysis into the neural networks that we have developed on Pentium-based PCs. With these additional features, we hope to develop a

flexible, robust, and easy-to-use system which is appropriate for population pharmacokinetic data analysis.

References and Notes

1. Yuh, L.; Beal, S.; Davidian, M.; Harrison, F.; Hester, A.; Kowalski, K.; Vonesh, E.; Wolfinger, R. *Biometrics* **1994**, *50*, 566–575.
2. Sheiner, L. B.; Ludden, T. M. *Annu. Rev. Pharmacol. Toxicol.* **1992**, *32*, 185–209.
3. Whiting, B.; Kelman, A. W.; Grevel, J. *Clin. Pharmacokinet.* **1986**, *11*, 387–401.
4. Veng-Pederson, P.; Modi, N. B. *J. Pharmacokinet. Biopharm.* **1992**, *20*, 397–412.
5. Hussain, A. S.; Johnson, R. D.; Vachharajani, N. N.; Ritschel, W. A. *Pharm. Res.* **1993**, *10*, 466–469.
6. Chow, H.; Chen, H.; Ng, T.; Myrdal, P.; Yalkowsky, S. H. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 723–728.
7. Gobburu, J. V. S.; Chen, E. P. *J. Pharm. Sci.* **1996**, *85*, 505–510.
8. Brier, E. M.; Zurada, J. M.; Aronoff, G. R. *Pharm. Res.* **1995**, *12*, 406–412.
9. Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. In *Parallel Distributed Processing*; Rumelhart, D. E., McClelland, J. L., the PDP Research Group, Eds.; MIT Press: Cambridge, MA, 1986; pp 45–76.
10. Sheiner, L. B.; Beal, S. L. *J. Pharmacokinet. Biopharm.* **1981**, *9*, 503–512.
11. Erb, R. J. *Pharm. Res.* **1993**, *10*, 165–170.
12. Rumelhart, D. E.; Widrow, B.; Lehr, M. A. *Commun. ACM* **1994**, *37*, 87–92.

Acknowledgments

This project was supported in part by a grant from the National Science Foundation (IRI9525790).

JS9604016