



Interactive Pharmacokinetics: A Software for Discovery, Analysis, and Simulation

İnteraktif Farmakokinetik: Keşif, Analiz ve Simülasyon Yazılımı

© Muhammed Cihan Güvel, © Canan Uluoğlu

Department of Medical Pharmacology, Gazi University Faculty of Medicine, Ankara, Türkiye

ABSTRACT

Objective: We aimed to develop “interactive pharmacokinetics (PK)”, an open-source, Python-based software package with a user-friendly graphical user interface. Our main goal was to overcome the cost and complexity barriers to industry-standard PK modeling tools and to facilitate the practical application of theoretical PK principles for students and researchers.

Methods: The software employs one, two, and three-compartment models, solving systems of differential equations to simulate drug concentration-time profiles. Parameter estimation is performed using non-linear optimization algorithms to minimize the sum of squared logarithmic errors. The platform is developed in Python, using NumPy and SciPy for core computations and the multiprocessing library for parallel processing of population data.

Results: A fully integrated, functional software application with a user-friendly graphical user interface was created. The platform enables users to run simulations and visualize results, including key parameters such as C_{max} and T_{max}. It performs *de novo* parameter estimation from patient data for one-compartment models and generates novel two- and three-compartment models using *a priori* data.

Conclusion: The software provides an accessible, no-cost tool for fundamental PK simulation and modeling. It serves as both a valuable pedagogical instrument for pharmacology education and a capable platform for preliminary research, particularly in resource-limited environments. Future work should focus on comprehensive validation against established software.

Keywords: Pharmacokinetic modelling, scientific software, drug development, clinical research

ÖZ

Amaç: Kullanıcı dostu bir grafiksel kullanıcı arayüzüne sahip, açık kaynak kodlu ve Python tabanlı bir yazılım paketi olan “interaktif farmakokinetik (FK)” geliştirmeyi amaçladık. Temel hedefimiz, endüstri standardı FK modelleme araçlarına erişimdeki maliyet ve karmaşıklık engellerini aşmak; öğrenciler ve araştırmacılar için teorik FK prensiplerinin pratik uygulamasını kolaylaştırmaktır.

Yöntemler: Yazılım, ilaç konsantrasyon-zaman profillerini simüle etmek için diferansiyel denklem sistemlerini çözerek bir, iki ve üç kompartmanlı modelleri kullanmaktadır. Parametre tahmini, logaritmik hata kareleri toplamını en aza indirmek için doğrusal olmayan optimizasyon algoritmaları kullanılarak gerçekleştirilir. Platform; temel hesaplamalar için NumPy ve SciPy kütüphanelerini, popülasyon verilerinin paralel işlenmesi için ise multiprocessing kütüphanesini kullanarak Python dilinde geliştirilmiştir.

Bulgular: Kullanıcı dostu bir grafiksel kullanıcı arayüzüne sahip, tam entegre ve işlevsel bir yazılım uygulaması oluşturuldu. Platform, kullanıcıların simülasyonlar yürütmesine ve C_{max} ile T_{max} gibi temel parametreler dahil olmak üzere sonuçları görselleştirmesine olanak tanımaktadır. Yazılım, tek kompartmanlı modeller için hasta verilerinden sıfırdan parametre tahmini gerçekleştirebilmekte ve önceden sağlanan verileri kullanarak yeni iki ve üç kompartmanlı modelleri oluşturabilmektedir.

Sonuç: Bu yazılım, temel FK simülasyonu ve modellemesi için erişilebilir ve ücretsiz bir araç sunmaktadır. Araç, özellikle kaynakların kısıtlı olduğu ortamlarda, hem farmakoloji eğitimi için değerli bir pedagojik enstrüman hem de ön araştırmalar için yetenekli bir platform görevi görmektedir. Gelecekteki çalışmalar, kabul görmüş yazılımlara karşı kapsamlı bir validasyon gerçekleştirilmesine odaklanmalıdır.

Anahtar Sözcükler: Farmakokinetik modelleme, bilimsel yazılım, ilaç geliştirme, klinik araştırma

Cite this article as: Güvel MC, Uluoğlu C. Interactive pharmacokinetics: a software for discovery, analysis, and simulation. Gazi Med J. [Epub Ahead of Print].

The preliminary results of this study have been accepted for oral presentation at the 28th National and 3rd International Congress of the Turkish Pharmacological Society, held in Antalya, Türkiye, from November 20-23, 2025.

Address for Correspondence/Yazışma Adresi: Muhammed Cihan Güvel, Department of Medical Pharmacology, Gazi University Faculty of Medicine, Ankara, Türkiye

E-mail / E-posta: guvelmd@gmail.com

ORCID ID: orcid.org/0000-0002-8097-0853

Received/Geliş Tarihi: 20.10.2025

Accepted/Kabul Tarihi: 11.11.2025

Epub: 10.12.2025



©Copyright 2025 The Author. Published by Galenos Publishing House on behalf of Gazi University Faculty of Medicine. Licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) International License.

©Telif Hakkı 2025 Yazar. Gazi Üniversitesi Tıp Fakültesi adına Galenos Yayınevi tarafından yayımlanmaktadır. Creative Commons Atıf-GayriTicari-Türetilemez 4.0 (CC BY-NC-ND) Uluslararası Lisansı ile lisanslanmaktadır.

INTRODUCTION

Pharmacokinetics (PK) is a fundamental discipline of pharmacology that studies the disposition of a drug in the body, encompassing the processes of absorption, distribution, metabolism, and excretion (ADME). PK modeling, in turn, utilizes a mathematical framework to define and predict these ADME processes. These models provide key information on a drug's efficacy and safety by predicting its time-concentration profile within the body *in silico* and are used to optimize therapeutic outcomes preclinically (1).

The model-based drug development (MBDD) approach, conducted through PK modeling, aims to reduce the high failure rates in clinical trials by providing better predictions about drug exposure-response relationships. Another important aspect of this approach is its acceptance not only by scientists but also by regulatory bodies. Leading regulatory agencies like the U.S. Food and drug administration and the European Medicines Agency actively encourage the use of MBDD and view it as a critical component of marketing authorization applications (2).

The growing importance of PK modeling has led to the development of various specialized software platforms designed to perform these analyses. This ecosystem consists of both commercial and open-source tools, each specializing in specific types of analysis and is adapted to different user profiles.

Non-linear mixed-effects modeling (NONMEM) is a software package that has long been considered the gold standard for population PK/pharmacodynamic analyses. Its primary function is to analyze typically sparse data from patient populations to investigate the sources of variability in drug response (3).

Physiologically based PK (PBPK) modeling is an approach that mechanistically predicts ADME properties based on fundamental physiological and biochemical principles. The *simcyp* simulator stands out as the most prominent platform in this field. It is known for its comprehensive libraries of diverse virtual populations and its ability to predict complex scenarios, such as drug-drug interactions (4).

Even though the current landscape of PK modeling software rests on a strong foundation, certain challenges and limitations present significant opportunities for developing new tools. These challenges include a lack of usability, accessibility, and integration with modern scientific computing platforms. All of these are directly linked to a broader issue encountered in pharmacology education. Students and young researchers find it difficult to translate theoretical principles into practical applications, a process known as “contextual learning transfer” (5). Software with complex interfaces deepens this educational gap and makes it difficult for the next generation of scientists to adopt these critical methods (6).

All the dominant platforms that are considered industry standards are commercial products. This restricts access for academic institutions, small biotechnology companies, and, in particular, researchers in low- and middle-income countries.

In this study, we developed an open-source, Python-based software package named “Interactive PKs” for PK modeling and simulation to address these challenges. Our goal is to provide a powerful experimental tool open to further development by experienced pharmacologists and an intuitive learning platform for students and researchers new to the field.

MATERIALS AND METHODS

The software developed in this study was designed as an integrated platform that combines two fundamental components of PK analysis—simulation and parameter optimization—on a modern computational infrastructure.

Libraries Used

The software is based on Python 3.13, chosen for its rich ecosystem of scientific libraries. Tkinter, Python's standard interface library, was used to provide a desktop user experience and to make the software easily accessible to a wide audience. Two critical functions at the core of the software—solving differential equations and non-linear optimization—are handled by the NumPy and SciPy libraries. The industry-standard Matplotlib library was used to generate simulation results and graphics.

The multiprocessing library was employed to accelerate computationally intensive tasks, such as parameter estimation from population data, via processor-level parallelism.

Simulation

The software used a compartmental modeling approach to simulate the drug concentration-time profiles in the body. One-, two-, and three-compartment models were represented as systems of differential equations describing the movement of the drug within the body. The equations used were based on differential pharmacological equations and on compartment-based PK models presented in Chapter 3 of Oğuz Kayaalp's Medical Pharmacology, 14th edition (7).

$$C(t) = \frac{fD}{v_d} \left(\frac{k_a}{k_a - k_e} \right) (e^{-k_e t} - e^{-k_a t})$$

Equation 1. Basic concentration-time equation

The concentration equation for the central compartment was linked to those of the peripheral compartments through the intercompartmental transfer constants k_{12} , k_{21} , k_{13} , and k_{31} . Parameters for the percentages of free and bound drug were determined for each compartment.

The compartmental relationships described by differential equations were simplified by reducing the number of compartments. A three-compartment model was created first, followed by two-compartment and single-compartment models.

The fundamental operating mechanism of the software is based on object-based and event-based programming, combined with solving differential equations. The software uses a numerical solver (`solve_ivp` from SciPy) to simulate the process over small time intervals.

The solver works by following these steps:

1. Starts at time 0: It knows the initial drug concentrations in each compartment.
2. Rate calculation: It uses differential equations to compute the instantaneous rate of change.
3. Steps forward: It takes a small “time step” (e.g., 1 minute) and determines the updated amount of drug in each compartment based on the calculated rates.
4. Repeats: Steps 2 and 3 are repeated until the end of the simulation period.

This process creates the concentration-time curve point by point. At the end of the process, an interactive concentration-time curve is displayed.

Parameter Estimation

In addition to simulations, the software includes parameter-estimation functions that predict key PK parameters [k_a , k_e , volume of distribution (V_d), etc.] from patient concentration-time data. This process is treated as a non-linear optimization problem.

The objective of the optimization algorithms is to minimize the difference between model-predicted concentrations (C_{pred}) and observed concentrations (C_{obs}). To achieve this, our software uses the sum of squared logarithmic errors as its primary function. This function is expressed mathematically in Equation 2.

$$Error = \sum (\log(C_{obs}) - \log(C_{pred}))^2$$

Equation 2. Sum of squared logarithmic errors

The aim of the optimization process is to find the model parameters that make the predicted curve match the actual patient data as closely as possible.

1. Initial Estimate: The process begins with an estimate of the parameters.
2. Simulation: Using these estimated parameters, a simulation is run to generate a predicted concentration curve.
3. Error Calculation (Loss Function): The predicted curve is compared with the actual patient data points to compute a single error (loss) value.
4. Adjust and Repeat: An optimization function (scipy.optimize.minimize) adjusts the parameter estimates to reduce the error. The gradient-based trust-constr and L-BFGS-B optimization algorithms are used to optimize this function. This “simulate-compare-adjust” loop is repeated until the error can no longer be reduced. The final parameters are those that produce the best-fitting curve.

When working with population data, calculating the error function for each individual can significantly slow processing. To overcome this computational bottleneck, our software leverages the multiprocessing library. This method allows for parallel computations on multiple processor cores. The goal of this approach is to make the analysis of large datasets more practical by significantly accelerating the optimization process compared with single-core processing.

Using a parameter-estimation approach, four functions were integrated into the software.

1. Creation of single-compartment parameters from real patient data: users can input patient concentration-time data to determine parameters for a one-compartment analysis.
2. Assessment of the fit of existing patient data to pre-existing two- or three-compartment model data: after entering patient concentration-time data, the system evaluates how well the data fit an a priori reference model.
3. New models are developed from existing patient concentration-time data using a pre-existing two- or three-compartment model: users enter the concentration-time data, and a new model is created by optimization against an a priori reference model.
4. Generation of two-compartment model parameters from a single-compartment model: unlike the first three methods, this function

uses both optimization and a heuristic approach to generate those parameters.

Attempt to Estimate Two-Compartment Model Data from One-Compartment Model Data

Attempting to generate parameters of a two-compartment model from one-compartment data represents the most experimental aspect of our software. For this purpose, a heuristic equation was developed.

$$k_{12} = \left(\frac{k_{e1comp}}{R^2} \right) * \left(\left(\frac{V_c}{V_p} \right) * K_p \right) * (e^{(R-1)*2.5})$$

Equation 3. Heuristic equation for estimating two-compartment parameters from single-compartment parameters. (*Equation Parameters:* K_{e1comp} : Elimination constant from the one-compartment model; V_c : Tissue compartment volume; V_p : Plasma compartment volume; V_d : Virtual distribution volume; K_p : Ratio of free drug ratio in plasma to free drug ratio in tissue; R : Ratio of V_d to ($V_c + V_p$); e : Euler's number).

This equation generates a two-compartment model from single-compartment data, using a formula derived from several assumptions. The known variables in this equation are manually entered values for V_d and k_{e1comp} the free drug fraction in plasma, and the calculated values for V_c and V_p based on body weight. The goal is to obtain an approximate value of k_{12} using these values.

We can explain this equation step-by-step as follows:

1. The calculation of $[(V_c/V_p) * K_p]$ in the middle of the equation involves dividing the amounts of free drug in the two compartments. The purpose of this calculation is to determine approximately whether the drug is more likely to move from plasma to tissue or from tissue to plasma. At steady state, we obtain the ratio of the total drug amount in the tissue compartment to that in the central compartment. This combined ratio is used to establish a balance between k_{12} (the rate of transfer from plasma to tissue) and k_{21} (the rate of return from tissue to plasma). The K_p value, which is calculated based on the drug-binding rate in tissue, is derived from the equation $V_d = V_p + [V_c * (\text{plasma free drug fraction/tissue free drug fraction})]$. This formula is based on the assumption that “as the free drug fraction in plasma increases and the free fraction in tissue decreases, the drug is expected to bind more extensively in tissue, thereby increasing the V_d .”
2. The value of R is calculated by dividing the virtual V_d , already known from the single-compartment model, by the total physiological volume ($V_p + V_c$), which is calculated from body weight. This ratio serves as an approximate measure of how much of a drug is retained in the tissues. For example, if the ratio is 5, it means the drug behaves as if it's distributed in a volume five times the physical volume. Therefore, the calculation $e^{(R-1)*2.5}$ suggests that, as R exceeds 1, the accumulation rate in the tissue will increase and k_{12} will increase proportionally as well.
3. However, based on the k_{e1comp}/R^2 calculation in the equation, the drug's elimination must decrease in proportion to R^2 . The drug's tendency to remain in the plasma decreases as R increases. Making k_{12} proportional to k_{e1comp} in this equation ensures the consistency of the two-compartment model. Assuming that a rapidly eliminated drug also distributes rapidly and that a slowly eliminated

drug distributes slowly, a more balanced model can be created. If there were no such relationship between k_{12} and k_{10} , it would be possible to produce strange and illogical models in which a drug is eliminated very quickly but enters tissues extremely slowly.

This equation is merely a heuristic estimate and cannot directly simulate real-life situations. We observed that attempts to use this approach were unsuccessful. Two-compartment model predictions generated by this equation were unsuccessful.

Subsequently, the curve estimated by this equation was considered a hypothetical model. The single-compartment and the hypothetical two-compartment model curves were optimized relative to one another. When an optimization approach was applied to the two curves, substantially more successful estimation curves were obtained. The partial success achieved at this stage was retained in the software as a basis for further experimentation and development. This is currently at an experimental stage and requires further mathematical development and testing in a real-world environment.

Finally, all the simulation and parameter estimation functions within the software have been integrated and presented to the user through a user-friendly interface.

This research focuses on the design and implementation of an open-source computational tool for PK modeling. The study methodology relies entirely on mathematical algorithms and theoretical data processing. As no human subjects, animal models, or biological samples were utilized or collected during the development and testing of this software, no ethical approval was sought or required for this work.

RESULTS

Based on the approaches applied, an integrated environment with a user-friendly interface was created. When the program is launched, the screen shown in Figure 1 appears.

The first feature on the interface is a control for selecting the number of compartments. Users can choose among one-, two-, and three-compartment models (Figure 1a). This is followed by the section for entering model parameters (Figure 1b). For the single-compartment model, the required parameters are the virtual V_d , plasma protein binding rate, simulation duration, simulation time step (calculation frequency), and steady-state concentration tolerance, which determines the precision of the steady-state concentration. The model-parameter screen shows minor differences between the two- and three-compartment models. For instance, in the two-compartment model, the user must determine the transfer coefficients from plasma to tissue (k_{12}) and from tissue to plasma (k_{21}), and the percentage of protein binding in the tissue (Figure 2). Next, the user is prompted to select the route of administration, with options for oral, IV bolus, and IV infusion. Here, the user also determines whether elimination will follow first-order or zero-order kinetics (Figure 1c).

In the block below, the elimination parameters are entered according to the chosen elimination type. If first-order kinetics is selected, the elimination constant (k_e) is required; if zero-order kinetics is selected, the user must provide the V_{max} and K_m values (Figure 1d).

The next block, which is optional, is for a loading dose. If this option is

selected, the user is prompted to provide the loading dose (Figure 1e).

Finally, the user is prompted to specify the dose, number of doses, dosing interval, absorption rate, intestinal bioavailability, and hepatic bioavailability (Figure 1f).

After the “start simulation” button is clicked, a concentration-time graph will be generated (Figure 3). Additionally, the estimated half-life, T_{max} , C_{max} , clearance, steady-state concentration (C_{ss}), and time to reach the steady-state concentration will be calculated (Figure 1g).

Additionally, the software’s capabilities extend beyond mere visualization of differential equations.

De Novo Estimation Function from Patient Data

The software can create population data and estimate K_a and K_e using only patient blood concentration-time data and the provided volume of distribution and plasma protein-binding rates. Furthermore, it can generate concentration-time population curves based on these parameter estimates.

To use this function, the user inputs the patient number, dose, blood sampling time, and concentration values into the interface (Figure 4). The system then iteratively applies optimization algorithms. This process can take a relatively long time (30 minutes to 1 hour), depending on how close the initial estimates are to the correct values. Following this process, goodness-of-fit graphs can be generated at the user’s request. The operation concludes with the estimation of model parameters and the visualization of the predicted concentration–time profile based on these parameters (Figure 5).

Estimation from Patient Data: Two- and Three-Compartment Models

Because of the large number of unknown variables in the two- and three-compartment models, new values could only be obtained through optimization based on a previously prepared model. Therefore, an a priori model must first be entered into the system. The system then attempts to determine new parameter estimates for these patients using the patient concentration-time data provided to the software (Figure 6).

This method not only builds a new model from an a priori model but also evaluates how well the entered data fit the pre-existing predictive model.

Estimation from Single-Compartment Model Data to Two-Compartment Model Parameters

When the estimated curve resulting from the heuristic equation was treated as a hypothetical model, the single-compartment and hypothetical two-compartment model curves were optimized relative to each other. It was observed that optimizing the fit between the two curves produced substantially more accurate prediction curves than the initial attempt (Figure 7). Here, the criterion for success was the similarity between the curve of the single-compartment model and the newly created optimization curve; no mathematical similarity analysis was conducted. This point is still in the experimental stage and needs to be further developed mathematically and tested in a real-world environment.

Interactive Pharmacokinetics - #With Kinar CRO Contribution#

Font Size: [] [] []

One-Compartment Model Two-Compartment Model Three-Compartment Model

Volume of Distribution (Vd) (L): []

Plasma Protein Binding (%): []

Simulation Duration (hours): []

Time Step (minutes): []

Css Tolerance: []

Quick Predict Two-Compartment Model (Experimental) -- Estimate Two-Compartment Model via Optimization (Experimental) --

Zero-Start Estimation (Population Data)

Administration Route: oral

Elimination Type: ☒ Linear (First-Order) ☐ Michaelis-Menten (Zero-Order)

Elimination Constant (ke): []

Vmax (mg/hour): []

Km (mg/L): []

Apply Loading Dose

Maintenance Dose: Oral Parameters

Dose (mg): [] Absorption Rate ka: []

Number of Doses: [] Intestinal F (%): []

Interval (hours): [] Liver F (%): []

Run Simulation

Results

Half-Life: Infinite Clearance: -- Cmax: --

Tmax: -- Css: -- Tss: --

Vd: --

Figure 1. Main User Interface. (a) Selection of the number of compartments. (b) Model parameters. (c) Selection of the drug administration route and elimination type. (d) Determination of elimination parameters. (e) Selection of a loading dose. (f) Dose parameters, absorption rate, and bioavailability. (g) Calculated parameters.

One-Compartment Model Two-Compartment Model Three-Compartment Model

Model Parameters

Plasma Volume (Vp) (L): []

Plasma->Tissue k_{12} : []

Plasma Protein Binding (%): []

Simulation Duration (hours): []

Css Tolerance: []

Tissue Volume (Vt) (L): []

Tissue->Plasma k_{21} : []

Tissue Protein Binding (%): []

Time Step (minutes): []

Calculate Volumes Check Data Fit Estimate Parameters

Figure 2. View in a two-compartment model.

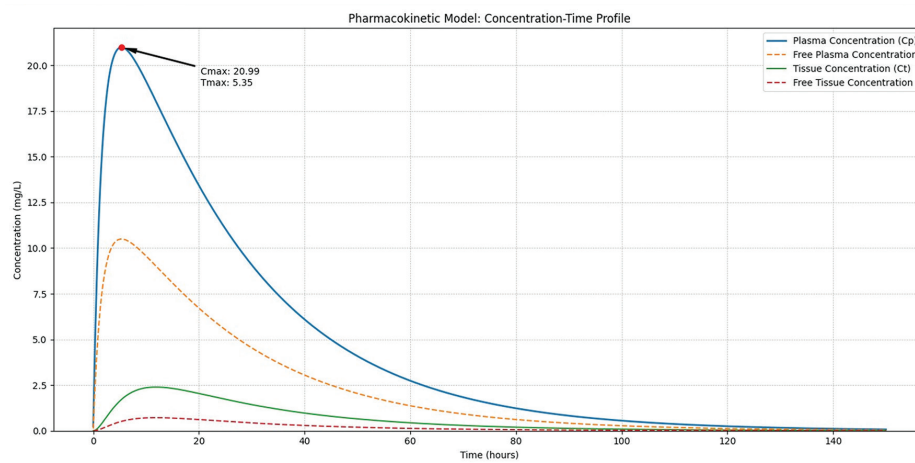


Figure 3. Concentration-time graph resulting from a two-compartment model calculation (the dark blue line: total plasma concentration. The light blue line: Free plasma concentration. The dark green line: Total tissue concentration. The light green line: free tissue concentration).

Zero-Start Estimation (Parallel Population Model)

Paste Data from Excel

1. Prepare your data in Excel in the following 4-column order:
 [Patient ID] [Dose (mg)] [Time (hours)] [Conc. (mg/L)]

2. WITHOUT header row, select only the cells containing data.
 3. Copy (Ctrl+C) and paste into the text box below (Ctrl+V).

P001	500	0.5	3.25
P001	500	1.8	6.40
P001	500	4.2	5.10
P001	500	8.3	2.45
P001	500	12.1	1.15
P002	500	0.8	4.80
P002	500	2.4	6.95
P002	500	3.9	5.85
P002	500	7.5	3.10
P002	500	11.5	1.45
P003	500	0.3	1.95
P003	500	1.5	5.90
P003	500	5.0	4.25

☐ Generate Goodness-of-Fit Plots

Estimate Parameters Using Pasted Data

Starting optimization for 4 patients...

Apply Estimated Parameters to Main Screen

Optimization Progress: Depending on data size, this may take from half an hour to several hours.

Trial count (Progress level): 4

Figure 4. *De novo* estimation function from patient data.

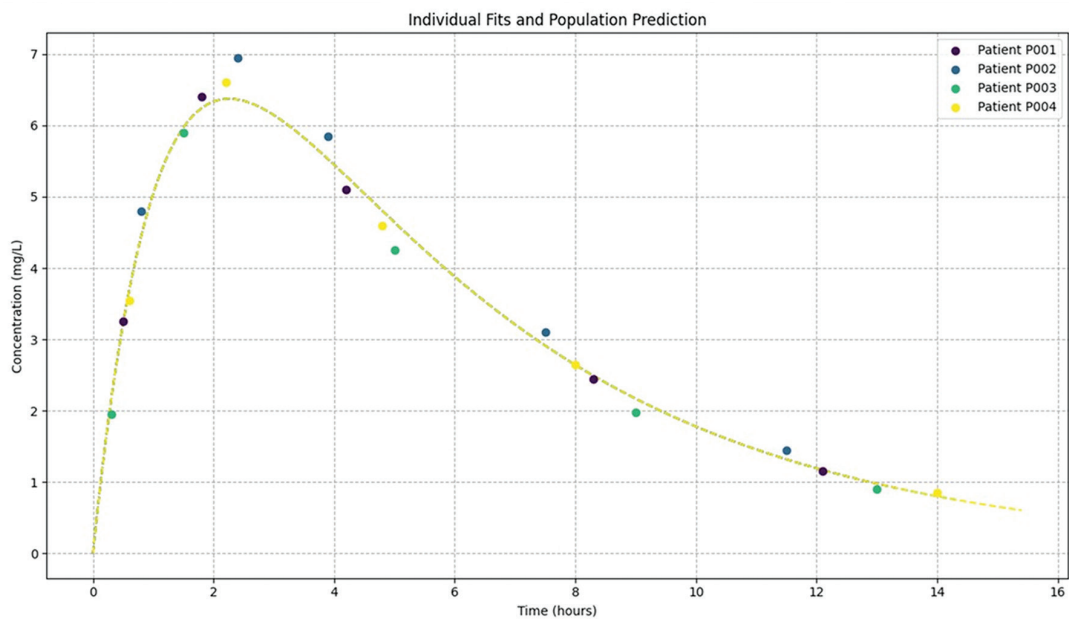


Figure 5. Visualization of the results from the *de novo* estimation function using patient data.

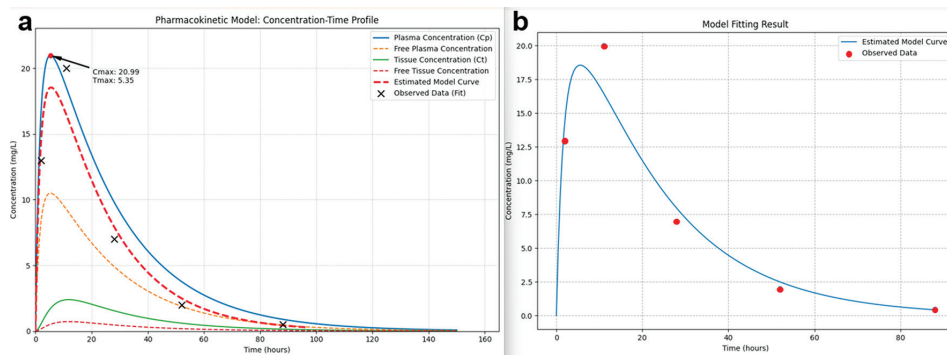


Figure 6. Visualization of the new concentration-time model estimated from patient data in a two-compartment model (a: The “x” marks show the patient data, while the dashed red line represents the new model curve created through optimization; b: The red dots indicate the new patient data entered, and the blue line shows the estimated model curve).

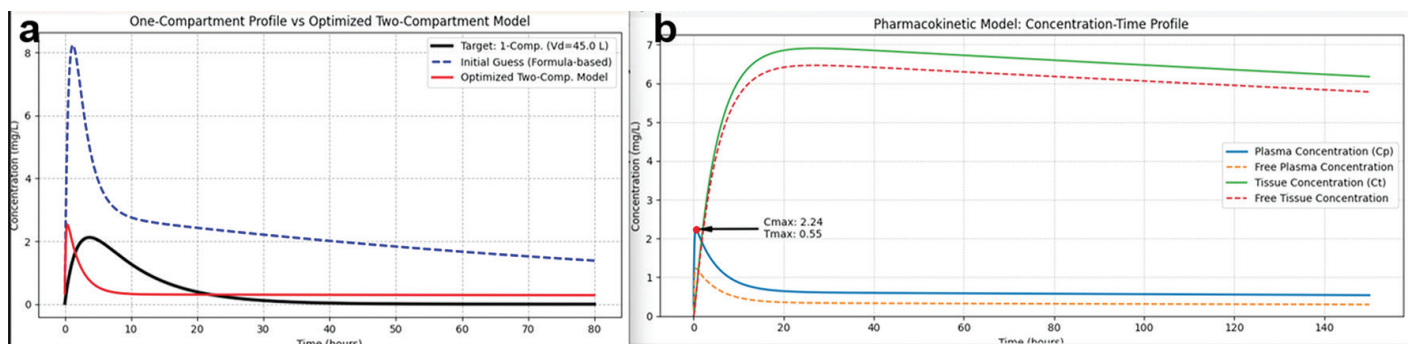


Figure 7. Estimation of two-compartment model parameters from single-compartment model data. (a) The dashed blue line represents the model resulting from the heuristic formula, the black line shows the single-compartment model, and the red line is the curve of the common model that emerged from optimization. (b) Panel b shows the new parameters created through optimization being tested in a two-compartment model simulation.

DISCUSSION

In this study, a software tool was developed to simulate and predict fundamental PK analyses. This software platform was designed to meet the needs of both educational and research fields.

The software interface is the most crucial element for ensuring accessibility and ease of use, which is our claim. In this context, the software's features should be regarded as a curated set of the most fundamental and frequently used tools in PKs. This highlights our strategic approach during the design phase. Rather than competing with the highly specialized and comprehensive features of commercial giants, our study focused on addressing the core curriculum of PK education and the fundamental needs of exploratory research. Therefore, the success of this work should be evaluated not by “what it can do”, but by “whom it empowers”.

Historically, the field of pharmacometric modeling has been dominated by powerful yet costly commercial software such as NONMEM and Simcyp. While these tools are considered industry standards, they create significant financial barriers. This makes them largely inaccessible to academic institutions, researchers in low-income countries, and small-scale companies. The open-source, free nature of the developed software is intended to completely eliminate this financial obstacle.

A second obstacle is the complexity and steep learning curve. Tools such as NONMEM are command-line-based and require in-depth knowledge of specialized programming languages or complex

control files for effective use (8). In contrast, our software's user-friendly graphical interface removes this barrier, allowing users to focus on PK principles without struggling with the software.

However, strong alternatives exist within the open-source domain. The most prominent of these is the nlmixr2 package, which is based on the R programming language. nlmixr2 is a code-based solution that requires programming proficiency (9). Additionally, user-interface add-ons like shinyMixR, which have been developed for this package, indicate a need for user-friendly platforms within this ecosystem (10).

Basing this software on Python places it in a unique position regarding its linguistic foundation. While R has deep roots in statistics and bioinformatics, Python is increasingly the dominant language for scientific computing, data science, and machine learning. Its larger talent pool and ability to integrate more smoothly into large-scale corporate or web-based workflows make Python an attractive choice for modern scientific software development (11). Consequently, this software is positioned not only as a PK tool but also as a potential component of a broader Python-based scientific workflow.

The software's value should also be assessed by its potential to transform how PKs is taught and applied. Its accessibility and ease of use have the potential to foster pedagogical innovation in education and to provide equal opportunities for researchers. The software transforms abstract equations into an interactive visual experience. It allows learners to change parameters such as dose,

clearance, and volume of distribution and instantly see the results on the concentration-time curve, providing a dynamic, intuitive way to learn how these concepts are interrelated. This type of hands-on, simulation-based learning is a powerful active learning strategy that can increase understanding and retention more effectively than static textbook examples. Therefore, it can be confidently stated that this tool can make the learning process more engaging and effective by allowing students to explore theory in a gamified manner and to observe the results immediately.

Study Limitations

The software is more than a simulator; it is also a predictive tool that incorporates several optimization functions, some of which are more effective, while others remain experimental. At this point, it is important to highlight the effort required to calculate the transition from a single-compartment model to a two-compartment model. Initially, a heuristic equation-based approach was attempted but proved unsuccessful. This failure is, in itself, a significant finding because it provides evidence of the non-linear nature of the relationship between single-compartment and two-compartment model parameters. While the second approach, based on optimization between two curves, was more successful, this feature is not yet mature.

On the other hand, the software's current capabilities must be evaluated within certain limitations. One of the most significant limitations is its focus on standard compartmental models in its current form. This excludes more complex and mechanistic modeling paradigms, such as PBPKs, which play a central role in modern drug development. This situation defines the boundaries of the software's current applicability.

Another critical limitation is the lack of a comprehensive official validation study. While the software is functional, its outputs have not yet been compared with those of established software or with standard reference datasets accepted by regulatory agencies. This step is indispensable for the software to be accepted as a reliable research tool. This validation is essential for adoption of the software by the scientific community and should be the top development priority for future work.

CONCLUSION

In this study, new open-source PK-modeling software built within the Python ecosystem was developed. The platform offers a package of fundamental simulation and parameter estimation tools within an accessible graphical user interface.

The primary and most novel contribution of this work is not the invention of new modeling algorithms. Instead, existing numerical methods have been holistically integrated into a platform that aims to overcome critical barriers—such as cost, complexity, and usability—that limit access to industry-standard tools. This approach is intended to make PK modeling capabilities accessible to a wider audience.

In doing so, the study makes a dual contribution to the field: on the one hand, it provides a powerful pedagogical tool to bridge the gap between theory and practice in pharmacology education, and on the other hand, it empowers researchers in resource-limited environments with a capable platform for preliminary modeling and hypothesis testing. We also believe that our work will make

significant contributions to postgraduate and lifelong learning in pharmacology. These capabilities require further improvement and validation through future development efforts.

Ethics

Ethics Committee Approval: This research focuses on the design and implementation of an open-source computational tool for pharmacokinetic modeling. The study methodology relies entirely on mathematical algorithms and theoretical data processing. As no human subjects, animal models, or biological samples were utilized or collected during the development and testing of this software, no ethical approval was sought or required for this work.

Informed Consent: Not required for this study.

Footnotes

Authorship Contributions

Surgical and Medical Practices: M.C.G., C.U., Concept: M.C.G., C.U., Design: M.C.G., C.U., Data Collection or Processing: M.C.G., C.U., Analysis or Interpretation: M.C.G., C.U., Literature Search: M.C.G., C.U., Writing: M.C.G., C.U.

Conflict of Interest: No conflict of interest was declared by the authors.

Financial Disclosure: The authors declared that this study received no financial support.

REFERENCES

- Ahmad AM. Recent advances in pharmacokinetic modeling. *Biopharm Drug Dispos.* 2007; 28: 135-43.
- Kimko H, Pinheiro J. Model-based clinical drug development in the past, present and future: a commentary. *Br J Clin Pharmacol.* 2015; 79: 108-16.
- Park WS. Pharmacometric models simulation using NONMEM, Berkeley Madonna and R. *Transl Clin Pharmacol.* 2017; 25: 125-33.
- Jamei M, Marciniak S, Feng K, Barnett A, Tucker G, Rostami-Hodjegan A. The simcyp population-based ADME simulator. *Expert Opin Drug Metab Toxicol.* 2009; 5: 211-23.
- Brackett C, Reuning R. Teaching pharmacokinetics using a student-centered, modified mastery-based approach. *Am J Pharm Educ.* 1999; 63.
- Fasinu PS, Wilborn TW. Pharmacology education in the medical curriculum: Challenges and opportunities for improvement. *Pharmacol Res Perspect.* 2024; 12: e1178.
- Onaran O. Temel farmakokinetik kavramlar. In: Kayaalp SO, Gümüşel B, Babaoğlu MÖ, Melli M, editors. *Oğuz Kayaalp Tıbbi Farmakoloji.* 14th ed. Ankara: Güneş Tıp Kitabevleri; 2024.
- Bauer RJ, Guzy S, Ng C. A survey of population analysis methods and software for complex pharmacokinetic and pharmacodynamic models with examples. *AAPS J.* 2007; 9: E60-E83.
- Fidler M, Wilkins JJ, Hooijmaijers R, Post TM, Schoemaker R, Trame MN, et al. Non-linear mixed-effects model development and simulation using nlmixr and related r open-source packages. *CPT Pharmacometrics Syst Pharmacol.* 2019; 8: 621-33.
- Hooijmaijers R. Introduction to the shinyMixR package [Internet]. CRAN; Available from: https://cran.r-project.org/web/packages/shinyMixR/vignettes/getting_started.html Accessed 2025 Oct 11.
- Pittard WS, Li S. The essential toolbox of data science: Python, R, Git, and Docker. *Methods Mol Biol.* 2020; 2104: 265-311.