Journal of Pharmacokinetics and Pharmacodynamics (2024) 51:187–197
https://doi.org/10.1007/s10928-024-09921-y

ORIGINAL PAPER

Evaluation of ChatGPT and Gemini large language models for pharmacometrics with NONMEM

Euibeom Shin[1] · Yifan Yu[1] · Robert R. Bies[1] · Murali Ramanathan[1]

Received: 29 March 2024 / Accepted: 16 April 2024 / Published online: 24 April 2024

Abstract

To assess ChatGPT 4.0 (ChatGPT) and Gemini Ultra 1.0 (Gemini) large language models on NONMEM coding tasks relevant to pharmacometrics and clinical pharmacology. ChatGPT and Gemini were assessed on tasks mimicking real-world applications of NONMEM. The tasks ranged from providing a curriculum for learning NONMEM, an overview of NONMEM code structure to generating code. Prompts in lay language to elicit NONMEM code for a linear pharmacokinetic (PK) model with oral administration and a more complex model with two parallel first-order absorption mechanisms were investigated. Reproducibility and the impact of "temperature" hyperparameter settings were assessed. The code was reviewed by two NONMEM experts. ChatGPT and Gemini provided NONMEM curriculum structures combining foundational knowledge with advanced concepts (e.g., covariate modeling and Bayesian approaches) and practical skills including NONMEM code structure and syntax. ChatGPT provided an informative summary of the NONMEM control stream structure and outlined the key NONMEM Translator (NM-TRAN) records needed. ChatGPT and Gemini were able to generate code blocks for the NONMEM control stream from the lay language prompts for the two coding tasks. The control streams contained focal structural and syntax errors that required revision before they could be executed without errors and warnings. The code output from ChatGPT and Gemini was not reproducible, and varying the temperature hyperparameter did not reduce the errors and omissions substantively. Large language models may be useful in pharmacometrics for efficiently generating an initial coding template for modeling projects. However, the output can contain errors and omissions that require correction.

Keywords Pharmacometrics · ChatGPT · Pharmacokinetics · Drug Development · Artificial intelligence · Generative AI · Modeling · Nonlinear mixed effects · NONMEM

Introduction

Large language models (LLM), as exemplified by ChatGPT from OpenAI, Gemini from Google, Llama from Meta, and Claude from Anthropic [1–5], are widely viewed as an important advancement in the field of artificial intelligence (AI) as they are capable of emulating human-like text generation and comprehension [6]. Usage of LLM has increased rapidly since they are versatile, user-friendly, and can assist with diverse tasks, e.g., engaging in casual conversations and document editing to solving complex, problem-oriented queries [7]. LLM can generate software code and language-based text with a degree of sophistication that suggests potential utility in pharmacometrics [8].

Pharmacometrics approaches enable analysis of the time courses and variability of drug concentrations [9], and are leveraged to inform dosage recommendations and therapeutic strategies [10]. NONMEM is a software package for implementing nonlinear mixed effects regression methods that is widely used by pharmacometricians [11]. NONMEM employs the NONMEM Translator (NM-TRAN) language for coding, which we refer to hereinafter simply as NONMEM code. NONMEM-coded nonlinear mixed effects regression analyses of compartmental pharmacokinetic models are commonly summarized in the new drug applications that are submitted to regulatory agencies such as the Food and Drug Administration and European Medicines Agency to support marketing approval for innovator products [12].

ChatGPT, Gemini, and other LLM have already proven useful for generating code in commonly used programming

✉ Murali Ramanathan
Murali@Buffalo.Edu

[1] Department of Pharmaceutical Sciences, University at Buffalo, The State University of New York, Buffalo, NY 14214-8033, USA

languages such as C++, Python, and R. Thus, integration of ChatGPT into a NONMEM workflow presents an emerging opportunity to enhance the productivity of coding tasks for pharmacometrics modeling [13, 14]. In addition to coding, ChatGPT could potentially streamline aspects of modeling documentation and be used for learning NONMEM and pharmacometrics.

The key research aim was to assess the capabilities and limitations of ChatGPT 4.0 and Gemini Ultra 1.0 in interpreting lay language prompts for pharmacometric modeling tasks designed to elicit corresponding NONMEM code. The tasks were selected to mirror the practical applications of NONMEM in pharmacometrics and clinical pharmacology settings [10] and covered a range of activities, from providing a curriculum for learning and understanding NONMEM code structure to developing code for different pharmacokinetic (PK) models. The NONMEM tasks included coding for a linear PK model with oral administration and a more complex one-compartment model with two parallel first-order absorption mechanisms.

Methods

ChatGPT & Gemini methods

The default version of ChatGPT 4.0 [15] and Gemini Ultra 1.0 [2] were run at chat.openai.com and gemini.google.com, respectively, on a MacBook Air computer running macOS Ventura 13.5.1. Screenshots from individual runs were saved.

ChatGPT 4.0 and Gemini Ultra 1.0 are hereinafter referred to as ChatGPT and Gemini, respectively.

Case studies

Two types of prompts were provided to ChatGPT and Gemini to generate output. The first related to teaching elements of NONMEM (Case Studies 1 and 2) and the second related to generating NONMEM code for a specific population PK example (Case Studies 3A and 3B).

We assessed ChatGPT and Gemini's capability to comprehend and instruct on pharmacokinetic/pharmacodynamic (PK/PD) modeling with NONMEM, focusing on a one-compartment model analysis.

A structured sequence of prompts was designed to engage the LLM in a step-by-step process, beginning with an explanation of NONMEM principles and culminating in a practical demonstration of NONMEM code. For convenience of presentation, Case Study 1 was a curricular framework for learning NONMEM in context, Case Study 2 was a request for a specific explanation of NONMEM code, whereas Case Study 3A and Case Study 3B evaluated the ability of ChatGPT and Gemini to generate NONMEM code for specific pharmacokinetic models. The following prompts were entered for the four Case Studies.

Case study 1

If you were a teacher, what topics related to NONMEM would you teach students?

Case study 2

You are an expert NONMEM code writer for population PKPD in the pharmaceutical industry. You are always accurate and precise when formulating NONMEM code. Please describe the structure of NONMEM code using a basic example.

Case study 3A

Can you provide the NONMEM code for a linear PK model with oral administration?

Case study 3B

Provide NONMEM code for a one-compartment model with two parallel absorptions and linear elimination.

Case 1, Case 2, and Case 3A were run together in a single chat session, and Case 3B was running on a separate chat.

The prompt engineering strategy included persona crafting to embody an expert NONMEM coder, role assignment to align with instructional tasks, and sequential prompting to maintain context throughout the interaction.

The code generated was reviewed carefully for errors and omissions by two individuals (authors: YY, RRB) with NONMEM expertise.

Evaluation of reproducibility

Three replicate experiments were conducted with ChatGPT and Gemini for Case Study 3A and Case Study 3B. The prompts described in the preceding section were re-used verbatim.

Evaluating the effect of varying the "Temperature" hyperparameter

The "temperature" hyperparameter (T) of ChatGPT controls the balance between randomness and determinism in the results. Experiments with T values of 0.1, 0.33 and 1 were conducted for Case Study 3A and Case Study 3B.

These runs were conducted using OpenAI Playground, Model: GPT-4 Turbo preview (https://platform.openai.com/docs/models), which allows the temperature parameter

to be set by the user. The other parameters were fixed to *Maximumlength* = 4095, *TopP* = 1, *Frequencypenalty* = 0, and *Presencepenalty* = 0.

Results

Case study 1: Curricular framework for teaching NONMEM

Case study 1

If you were a teacher, what topics related to NONMEM would you teach students?

Figure 1 compares NONMEM related topics provided by ChatGPT and Gemini results.

ChatGPT

The ChatGPT's response reflects its utility for structuring a NONMEM curriculum that combines foundational knowledge with advanced concepts and practical skills.

The topic list proposed by ChatGPT starts with pharmacometrics principles to establish a foundation before progressing to advanced aspects. It provided technical requirements for NONMEM software installation, setup, and data management.

The response also included a detailed exploration of NONMEM code structure and syntax to equip learners with coding skills. Advanced topics, like covariate modeling and Bayesian approaches were integrated and showcasing the curriculum's depth. Practical applications, including simulations and predictive model checks, were also covered.

Moreover, the topic list addressed regulatory considerations and documentation practices in PKPD modeling, aligning with industry standards. It also keeps pace with current trends and future directions in pharmacometrics, ensuring contemporary relevance.

Gemini

The overall outline of Gemini's syllabus resembled the ChatGPT's in that it listed the introduction to PKPD, the fundamental structure of the NONMEM, basic structural PK models, and intermediate/advanced topics.

Gemini suggested learning foundational concepts of pharmacometrics and NONMEM including inter-individual variability and covariate modeling and then covering topics related to the fundamentals of more intermediate/advanced population agnostic pharmacokinetic models such as Michaelis–Menten elimination and target-mediated drug disposition. However, inter-individual variability, a basic topic was included in the intermediate section. It further provided the model evaluation and diagnostic methods, including the visual predictive checks and bootstrap techniques. Gemini included some basic goodness of fit approaches under an advanced topics heading. Gemini ended with a similar emphasis as ChatGPT, by analyzing real-world PKPD datasets.

Case study 2. Explanation of NONMEM code

Case study 2

You are an expert NONMEM code writer for population PKPD in the pharmaceutical industry. You are always accurate and precise when formulating NONMEM code. Please describe the structure of NONMEM code using a basic example.

Figure 2 summarizes the ChatGPT and Gemini results for Case Study 2.

ChatGPT

ChatGPT effectively communicated complex pharmacometrics concepts and provided an informative summary of NONMEM code structure.

The key components of NONMEM code, starting with the Problem Statement (\$*PROB*), which functions as the model's title or description, were provided. This was followed by Input Data Specification (\$*DATA*) for defining the dataset file and format, and Input Variable Definitions (\$*INPUT*), detailing essential variables such as *ID*, *TIME*, *DV*, and *AMT*.

The Model Specification (\$*PRED* or \$*MODEL*) section containing model equations, Estimation Method (\$*ESTIMATION*) for the selecting parameter estimation algorithm, and Output Specification (\$*TABLE*) for defining the output format were also described. The response also included sections such as \$*COV* for specifying the covariance matrix and \$*THETA* for providing starting parameter estimates and bounds.

ChatGPT also provided a basic pharmacokinetic model as an example (Fig. 2), clarifying key parameters such as clearance (*CL*), volume of distribution (*V*), and absorption rate constant (*KA*).

Figure 2 also summarizes the errors in the model components. First, ChatGPT incorrectly used \$*PARAM* to define initial parameter estimates instead of \$*THETA*, which is the appropriate block for defining initial estimates for fixed effects parameters. Secondly, ChatGPT provided an incorrect example for \$*PRED*. ChatGPT used \$*PRED* to code a one compartment model with first order absorption, without using *PREDPP*, a library of pre-written modeling in NONMEM. However, it incorrectly stated that \$*PRED* is for "predictive model", and the equations presented by ChatGPT within \$*PRED* were incorrect. For the one-compartment, first-order absorption

Fig. 1 Topics related to NONMEM identified by ChatGPT and Gemini

model, the Bateman function should be used to calculate concentration in the central compartment:

$$Y = AMT * KA/(V * KA - CL/V) * (EXP(-CL/V * TIME) - EXP(-KA * TIME))$$

Moreover, In the final part of ChatGPT's answer, it mistakenly identified $*COV* and $*THETA* as "Control Termination".

Additionally, ChatGPT did not provide any introduction for $*PK* or $*SUBROUTINE* which are important code components for modeling the values of basic and additional PK parameters when using *PREDPP*.

Fig. 2 The ChatGPT prompt and the resulting output from ChatGPT for Case Study 2. The errors and omissions in the code are highlighted in rectangles and described in the gold boxes

Gemini

Gemini provided the core structures of the NOMEM's NM-TRAN control records at the beginning and the code for a simple one-compartment PK model. The structure of NONMEM code was listed in the expected order: problem statement (*$PROBLEM*), data input (*$INPUT*, *$DATA*), model subroutine selection (*$SUBROUTINE*),

PK parameters ($*PK*), defining the residual error ($*ERROR*) and outputting the results ($*TABLE*). However, the simple one-compartment PK example provided was missing the crucial initial estimate code blocks: initial estimates for the OMEGA matrix ($*OMEGA*), initial estimates and bounds of fixed effects parameters ($*THETA*), and initial estimates for the NONMEM SIGMA matrix ($*SIGMA*). Additionally, no inter-individual variability was specified in the $PK block (i.e., the THETA values were not associated with a function that linked them to the inter-individual variability, e.g., $THETA(1) * exp(ETA(1))$ etc.

Case study 3. NONMEM coding

Case study 3A

Can you provide the NONMEM code for a linear PK model with oral administration?

Figure 3 summarizes the ChatGPT and Gemini results for Case Study 3A.

ChatGPT

ChatGPT generated an instructive example of NONMEM code for a basic linear PK model with oral administration.

Fig. 3 Results from ChatGPT (left) and Gemini (right) for the linear pharmacokinetic compartmental model with oral absorption (Case Study 3A). The ChatGPT and Gemini explanation of key components is cropped to improve readability of the figure. The errors and omissions in the code are highlighted in rectangles and described in the gold boxes

The response begins with the $*PROB* statement, effectively setting the context as a basic linear PK model for oral drug administration. The inclusion of $*DATA* and $*INPUT* sections, specifying the datafile and defining essential data columns like *ID*, *TIME*, *AMT*, and *DV*.

In the $*SUBROUTINES* section, ChatGPT appropriately selected *ADVAN*2 and *TRANS*2, suitable for a one-compartment model with first-order absorption and elimination. The $*PK* section, which is used to model the values of basic and additional PK parameters, was well-constructed, with clear definitions for the absorption rate constant (*KA*), clearance (*CL*), and volume of distribution (*V*), along with the calculation for the elimination rate constant (*K*).

ChatGPT's choice of the first order conditional estimation with interaction (FOCE INTERACTION) method in the $*ESTIMATION* section, and the request for covariance analysis ($*COVARIANCE*) and output specifications ($*TABLE*), demonstrates an "understanding" of the modeling process in NONMEM.

The $*ERROR* block incorrectly specified the residual error model. The formulas did not align with the standard additive or proportional error models and conflicted with the proportional error indicated in the $*SIGMA* block. For a proportional error model, the correct equation is:

$$Y = IPRED + IPRED * EPS(1)$$

The LLM's choice of the FOCE INTERACTION method in the $*ESTIMATION* section was problematic, leading to error messages upon running the code, necessitating a revision to $METHOD = 1$ or $METHOD = COND$. In the $*TABLE*, previously calculated elimination rate constant was not included in the output.

Several revisions made to address the errors in the control stream before NONMEM would run without errors and warnings. These are presented sequentially for illustrative purposes in the Supplementary File.

Gemini

Gemini generated a NONMEM control stream for a one-compartment model with first-order absorption, which is also provided in Case 2 as an example of the NONMEM code. The response provided reasonable $*PROBLEM* for the correct input item in $*INPUT* section.

In the $*SUBROUTINES*, Gemini appropriately selected *ADVAN*2 *TRANS*2 for one-compartment model with first order absorption. However, it also generated the $*MODEL* that cannot be used in the selected subroutine. This redundant code block will cause errors when running the model in NONMEM.

Gemini correctly selected *CL*, *V*, and *KA* as the PK parameters in $*PK*, and included exponential BSV on *CL* and *V*. However, incorrect parameter names (*TVCL* and *TVV*) were used. In addition, the scaling factor $S2 = V$ is missing in $*PK*.

The $*ESTIMATION* block is problematic as Gemini used $METHOD = FOCEI$ to specify *FOCEI* method, which should be coded as $METHOD = 1$ *INTERACTION* or $METHOD = COND\ INTERACTION$. Gemini also chose an inappropriate term and value for number of significant digits using $SIG = 0.2$ instead of an integer value for *NSIG*. Significant digits are specified when using the differential equation subroutines (where a solution tolerance is specified), not when using the algebraic solutions for simple PK models in PREDPP.

As for the output table, Gemini included *IPRED* and *DVRES*, which were not defined in the previous section. More importantly, Gemini missed $*THETA*, $*OMEGA*, and $*SIGMA* in the control stream, leaving the control stream incomplete. This error also occurred in Case 2.

Case study 3B

Provide NONMEM code for a one-compartment model with two parallel absorptions and linear elimination.

Figure 4 summarizes the ChatGPT and Gemini results for Case Study 3B.

ChatGPT

ChatGPT produced a NONMEM control stream for a one-compartment model with two parallel first order absorptions, which is a user-written model, that contained the correct code blocks. ChatGPT provided a correct $*PROBLEM* code block and defined the key data items correctly in the dataset in $*INPUT* block. However, in the $*DATA* code block, ChatGPT incorrectly included $IGNORE = CMT(3)$.

In the $SUBROUTINE section, ChatGPT inappropriately selected *ADVAN*2 and *TRANS2*, which generated an error. For the control stream using differential equations, *ADVAN*6, *ADVAN*8, *ADVAN*9 or *ADVAN*13 should be selected, and a relative tolerance *TOL* needed to be specified.

In the $*MODEL* section, ChatGPT correctly included two absorption depots and a central compartment. However, ChatGPT incorrectly defined the central compartment as the default dosing compartment.

In the $*PK* section, ChatGPT correctly defined two first-order absorption rate constants (*KA*1 and *KA*2), clearance (*CL*), and volume of distribution (*V*) along with their between-subject variability (*BSV*). The ChatGPT code failed to distribute the dose into two different absorption pathways because a fraction parameter was not included in $PK. The presence of redundant code regarding the *ETA* parameters also resulted in an error message. The $*DES* section had the correct ordinary differential equations for this model.

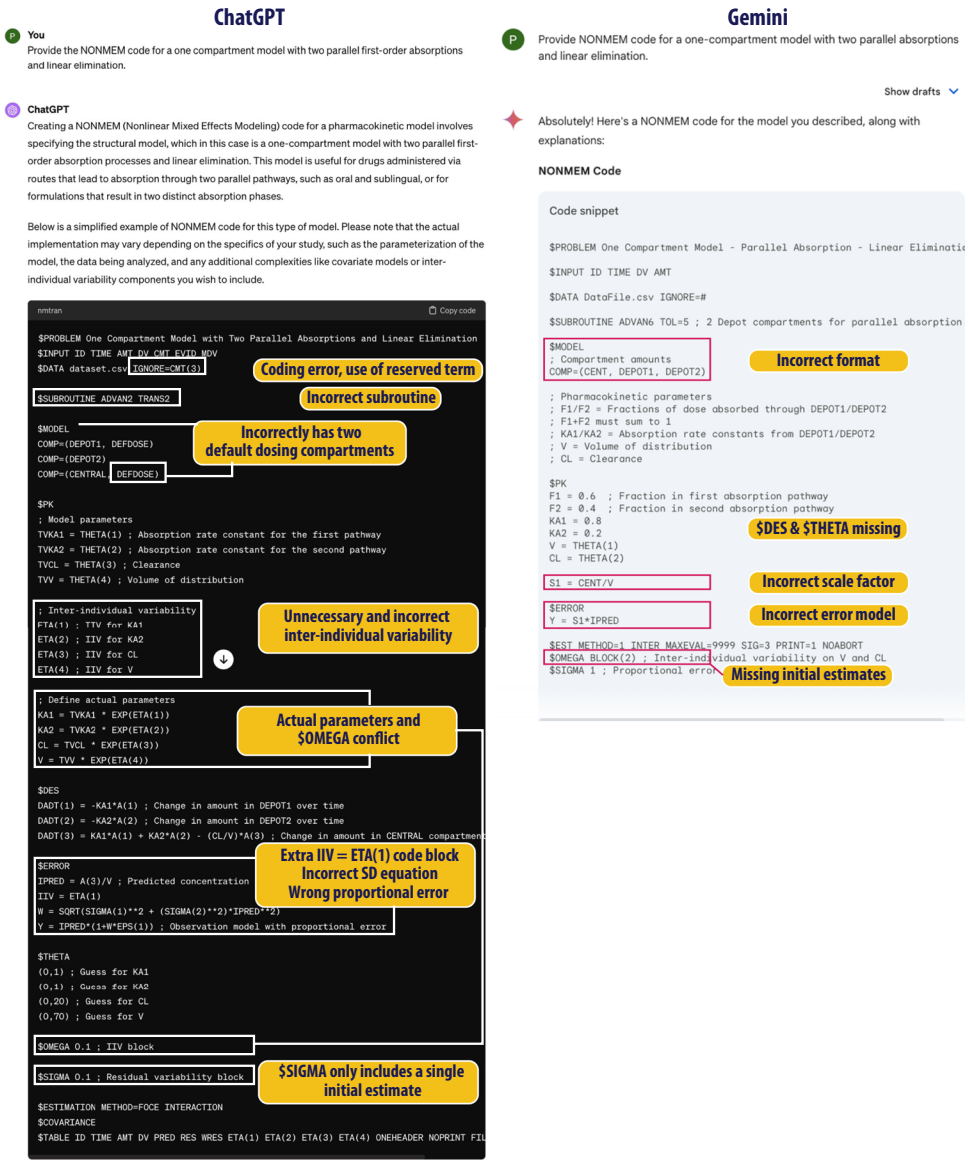

Fig. 4 Results from ChatGPT (left) and Gemini (right) for a one-compartment model with two parallel absorption pathways and linear elimination (Case Study 3B). The ChatGPT and Gemini explanation of key components is cropped to improve readability of the figure. The errors and omissions in the code are highlighted in rectangles and described in the gold boxes

Certain error patterns identified in Case Study 3A persisted in Case Study 3B, e.g., the $*ERROR* code block had incorrect standard deviation and proportional error model formulae. In addition, ChatGPT only provided a single initial estimate for the *OMEGA* matrix, and the code erroneously contained *METHOD* = *FOCE* to specify *FOCE* method instead of the correct *METHOD* = 1 or *METHOD* = *COND*.

The revisions made to address the errors and to obtain a NONMEM run without errors and warnings are summarized in the Supplementary File.

Gemini

Gemini provided a NONMEM control stream for a one-compartment model with parallel absorption and linear elimination. After generating a reasonable $*PROBLEM* block, it correctly coded $INPUT and $*DATA*, with required input items.

Gemini correctly selected *ADVAN*6 as the subroutine, as this user-written model required differential equation solver. It also provided a reasonable value for the relative tolerance *TOL*. However, the $*MODEL* generated by Gemini is problematic and should be revised to *COMP* = (*DEPOT*1) *COMP* = (*DEPOT*2) *COMP* = (*CENT*).

In $*PK*, Gemini correctly defined all the required PK parameters in this PK model, including *F*1, *F*2, *KA*1, *KA*2, *V*, and *CL*. However, it provided an incorrect scale factor for the central compartment and did not specify any inter-individual variability on the parameters.

The $*ERROR* block incorrectly specified the residual error model. A correct $*EST* was coded specifying FOCE method with interaction. Additionally, no initial estimates

were provided for the $*OMEGA* matrix. It's worth noting that Gemini missed including $*DES*, $*THETA*, and $*TABLE* in the control stream. These components are essential, and the model will not run without them.

Evaluation of reproducibility

The results from LLM can vary from one run to another for the same prompt. Three replicate experiments were conducted with ChatGPT and Gemini for Case Studies 3A and 3B.

ChatGPT

All three replicates contained errors and omissions. The results from the three replicate experiments for ChatGPT are summarized in Supplementary Figure S1 for Case Study 3A. In all three replicates: i) $*DES* and $*MODEL* were incorrectly included when using ADVAN2 and TRANS2, ii) the FOCE INTERACTION method in the $*ESTIMATION* section was not properly coded as $METHOD = 1$ or $METHOD = COND$., and iii) either the IIV and/or error model definitions were incorrect. Two of the replicates did not have the correct $S2 = V$ scaling factor. When included, all the $MODEL and $DES were either problematic or blank.

There were several errors unique to each replicate that occurred in different code blocks. Supplementary Figure S1 shows the corresponding replication experiments for Case Study 3A.

For Case Study 3B (Supplementary Figure S3), all three replicates contained the structural outline of the NONMEM control stream. However, in all three replicates, ChatGPT provided an initial estimate in $OMEGA without including between-subject variability on any parameters in $PK. The other errors were variable across the three trials (Supplementary Figure S3).

Gemini

For Case Study 3A, Gemini omitted $DATA and $SUBROUTINE in all replicates; $THETA and $ERROR were missed in two replicates. In all three replicates, $MODEL and $DES were problematic or blank. Two replicates contained incorrect comment statements (Supplementary Figure S2).

For Case Study 3B, two Gemini trials (Supplementary Figure S4) omitted important NM-TRAN control records such as $PROBLEM, $OMEGA, $TABLE, $DATA, and $SUBROUTINE. All three replicates were missing CMT in $INPUT. The second and third control streams used incorrect code in $PK. There other errors are highlighted (Supplementary Figure S4).

Effect of varying "Temperature" hyperparameter

The results from experiments with "temperature" hyperparameter (T) values of 0.1, 0.33 and 1, using ChatGPT for Case Study 3A and Case Study 3B, are summarized in Supplementary Figure S5 and S6, respectively.

For Case Study 3A, the code from all three temperature settings contained redundant $MODEL blocks and had incorrect scale factors in $PK. The $T = 0.1$ and $T = 0.33$ code correctly included between subject variability on all three PK parameters, whereas the $T = 1$ code inappropriately fixed all the PK parameters in $PK. The incorrect method name error wherein METHOD = FOCE was incorrectly used to specify the FOCE method, occurred only with $T = 0.33$.

For Case Study 3B, the $T = 0.1$ and $T = 0.33$ code inaccurately labeled default dose compartments. None of the trials correctly defined the fraction absorbed by the F1 and F2 pathways. The $T = 1$ code used atypical equations to define between subject variability on PK parameters that were found to be correct. Other common mistakes were also noted including incorrect method selection and conflicting IIV definitions.

Discussion

We evaluated ChatGPT and Gemini in several case studies that would enable us to delineate the strengths and limitations of using LLM for NONMEM coding, and to identify patterns in the coding errors that occurred. We focused on NONMEM because it is the most widely used coding framework in pharmacometrics. However, because most versions of NONMEM are marketed commercially and not freely available to open-source software users [16], the volume of NONMEM code readily accessible online is modest compared to more widely used languages such as C++, Python, and R. This makes NONMEM a challenging framework for LLMs, which depend on the availability of adequate training data.

The strength of ChatGPT and Gemini for NONMEM coding is that it can use the information from a short prompt provided by the user in lay language and provide a template of code containing the key components of a NONMEM control stream. The output from these LLMs is a good starting point for a coding project. The performance of ChatGPT and Gemini on the NONMEM coding tasks contained flaws and had focal structural (e.g., in Case Study 3B, the fraction of dose distributed between the parallel absorption compartments was not included) and NONMEM syntax (e.g., the FOCE method was incorrectly coded in both Case Study 3A and 3B) errors that require correction. We also found odd code snippets, e.g., NONMEM does not have $*PARAM* code block. We were surprised at the overall quality of the syllabus outline

generated by ChatGPT and Gemini in Case Study 1, given that pharmacometrics is a niche sub-specialty in the pharmaceutical industry, and there are only a few academic institutions that conduct cutting-edge research projects or provide training with NONMEM [17].

Between run variability is an intrinsic feature in all LLM including ChatGPT and Gemini because the underlying algorithms are based on generative AI methodologies [7, 18]. We therefore conducted reproducibility experiments to gauge the extent of the variability. Although the structural outlines from the replicate runs for Case Study 3A and Case Study 3B were generally similar, each instance of code produced had different errors. A frequent error in the ChatGPT NONMEM code was incorrect assignment of *METHOD = FOCE*, which recurred in 8/12 prompts in the replication and temperature hyperparameter experiments,

The "temperature" hyperparameter T, which ranges from 0 to 1 (default value = 0.7), controls the balance between the deterministic and generative aspects of ChatGPT. A low T value setting of 0.1 increases the likelihood of more deterministic and consistent outputs whereas a high T value setting of 1 increases the likelihood that more "creative" outputs with greater generative randomness will be produced. Interestingly, the code from the high $T = 1$ setting for individual parameters in the $PK block of Case Study 3B (Supplementary Figure S6) was atypical but correct. The temperature hyperparameter setting option was not publicly available in the Gemini Ultra 1.0 version that we used.

The code from both ChatGPTand Gemini are imperfect at generating NONMEM code. Shared errors patterns were seen across both LLMs, e.g., conflicting code blocks and selection of the incorrect method. However, each LLM had areas of strength and weakness. ChatGPT performed well in providing a structural outline of the control streams by providing required NM-TRAN control records that Gemini omitted ($SIGMA, $OMEGA, $THETA). Gemini's code was more carefully fine-tuned, and in Case 3B, it contained the fractions of dose absorbed through the two parallel absorption pathways and correct removal of column names, which ChatGPT missed.

A weakness of our study is that we did not compare the NONMEM coding performance of ChatGPT to other LLM other than Gemini, e.g., Llama and Claude [1, 3, 5, 15]. We also did not examine ChatGPT and Gemini in the context of other software tools such as *mrgsolve* [19], *nlmxr* [20, 21], the Mixtran used by Monolix [22], and STAN [23] that might be used by pharmacometricians.

Comparatively, ChatGPT, Gemini, and other LLM are more proficient at generating code and it is projected that the adoption of LLM could yield significant productivity improvements for programmers. Cloesmeijer et al. found that ChatGPT can be used to obtain PK parameters from the literature, code a population model and generate visualizations in R [24]. However, Cloesmeijer et al. indicated that the performance of ChatGPT for NONMEM was poor but did not present research results [24]. In previous work, we found that R code generated with ChatGPT is of satisfactory quality [18]. The accuracy of ChatGPT in calculations involving exponential, logarithmic and trigonometric functions, which are common in pharmacometrics, is only ~ 50% [18, 25, 26]. Thus, conducting numerical PK and pharmacometrics calculations directly within the ChatGPT interface should be avoided for now. The citations generated by earlier versions of ChatGPT (e.g., ChatGPT 3.5) often contained errors that were referred to as "artificial hallucinations", but this issue has been substantially improved with the new algorithm [18, 27, 28]. Similarly, we expect that new enhancements to LLM will quickly mitigate the high frequency of error rates in numerical calculations.

In conclusion, our results show that the utility of ChatGPT and Gemini for NONMEM coding might be metaphorically viewed as a cup that is currently half full. Although ChatGPT and Gemini may be useful for generating early versions of code, the resulting code requires careful review and changes before it can be run. However, LLMs are rapidly improving in their capabilities, and it might not be long before they are proficient at many NONMEM coding tasks.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10928-024-09921-y.

Author contributions Euibeom Shin – Data analysis, manuscript preparation.
Yifan Yu – Data analysis, manuscript preparation.
Robert Bies – Manuscript preparation.
Murali Ramanathan – Study concept and design, data analysis, manuscript preparation.

Funding This is unfunded research. Support from Grant MS190096 from the Department of Defense Multiple Sclerosis Research Program for the Office of the Congressionally Directed Medical Research Programs (CDMRP) to the Ramanathan laboratory is gratefully acknowledged.

Declarations

Financial conflicts Dr. Bies receives grant funding from NIDA, NIAID, NICHD, USAID, Bill and Melinda Gates Foundation and serves as a consultant for Advanced Biosciences Laboratories (NIAID). In the past he has served as a consultant for Lumos Biopharma through NGT Biopharma Consultants. Dr. Murali Ramanathan received research funding from the National Multiple Sclerosis Society, Department of Defense, National Science Foundation, and National Institute of Neurological Diseases and Stroke. He receives royalty from a self-published textbook.

Competing interests The authors declare no competing interests.

References

1. ChatGPT Version Jan 2024 (2024) https://chat.openai.com/. Accessed Mar 2024
2. Gemini Team, Anil R, Borgeaud S et al (2023) Gemini: A Family of Highly Capable Multimodal Models. arXiv:2312.11805. https://doi.org/10.48550/arXiv.2312.11805. Accessed Dec 01 2023. https://ui.adsabs.harvard.edu/abs/2023arXiv231211805G
3. Llama 2: open source, free for research and commercial use (2024) https://llama.meta.com/llama2/
4. Touvron H, Martin L, Stone K, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. 2023: arXiv:2307.09288. https://doi.org/10.48550/arXiv.2307.09288. Accessed Jul 01 2023. https://ui.adsabs.harvard.edu/abs/2023arXiv230709288T
5. Meet Claude (2024) https://www.anthropic.com/claude
6. Orru G, Piarulli A, Conversano C, Gemignani A (2023) Human-like problem-solving abilities in large language models using ChatGPT. Front Artif Intell 6:1199350. https://doi.org/10.3389/frai.2023.1199350
7. Roumeliotis KI, Tselikas ND (2023) ChatGPT and Open-AI Models: A Preliminary Review. Future Internet 15(6):192
8. Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training
9. Owen JS, Fiedler-Kelly J (2014) Introduction to population pharmacokinetic/pharmacodynamic analysis with nonlinear mixed effects models. Wiley
10. Pétricoul O, Cosson V, Fuseau E, Marchand M (2007) Population models for drug absorption and enterohepatic recycling. Pharmacometrics: the science of quantitative pharmacology 345–382
11. Bauer RJ (2019) NONMEM tutorial part I: description of commands and options, with simple examples of population analysis. CPT Pharmacometrics Syst Pharmacol 8(8):525–537. https://doi.org/10.1002/psp4.12404
12. Sun H, Fadiran EO, Jones CD et al (1999) Population pharmacokinetics: a regulatory perspective. Clin Pharmacokinet 37:41–58
13 Cloesmeijer ME, Janssen A, Koopman SF, Cnossen MH, Mathôt RA, consortium S, (2024) ChatGPT in pharmacometrics? Potential opportunities and limitations. British J Clin Pharmacol 90(1):360–365
14. Shin E, Ramanathan M (2024) Evaluation of prompt engineering strategies for pharmacokinetic data analysis with the ChatGPT large language model. J Pharmacokinet Pharmacodyn 51(2):101–108
15. Bard Large language model. 2023. https://bard.google.com
16. Fidler M, Wilkins JJ, Hooijmaijers R et al (2019) Nonlinear Mixed-Effects Model Development and Simulation Using nlmixr and Related R Open-Source Packages. CPT Pharmacometrics Syst Pharmacol 8(9):621–633. https://doi.org/10.1002/psp4.12445
17. Bonate PL, Barrett JS, Ait-Oudhia S et al (2023) Training the next generation of pharmacometric modelers: a multisector perspective. J Pharmacokinet Pharmacodyn. https://doi.org/10.1007/s10928-023-09878-4
18. Shin E, Ramanathan M (2023) Evaluation of prompt engineering strategies for pharmacokinetic data analysis with the ChatGPT large language model. J Pharmacokinet Pharmacodyn. https://doi.org/10.1007/s10928-023-09892-6
19. mrgsolve: Simulate from ODE-Based Models. R package version 1.4.1. Metrum Research Group; 2024. https://github.com/metrumresearchgroup/mrgsolve
20. Fidler M, Hooijmaijers R, Schoemaker R, Wilkins JJ, Xiong Y, Wang W (2021) R and nlmixr as a gateway between statistics and pharmacometrics. CPT Pharmacometrics Syst Pharmacol 10(4):283–285. https://doi.org/10.1002/psp4.12618
21. nlmixr: an R package for population PKPD modeling. 2019. https://nlmixrdevelopment.github.io/nlmixr/index.html
22. Anonymous. Monolix documentation. Lixoft-SimulationsPlus. Accessed March 13, 2024, 2024. https://monolix.lixoft.com/single-page/
23. Stan Reference Manual. NumFOCUS; 2011. https://mc-stan.org/docs/reference-manual/
24. Cloesmeijer ME, Janssen A, Koopman SF, Cnossen MH, Mathot RAA, Symphony consortium (2024) ChatGPT in pharmacometrics? Potential opportunities and limitations. Br J Clin Pharmacol 90(1):360–365. https://doi.org/10.1111/bcp.15895
25. Frieder S, Pinchetti L, Chevalier A et al (2023) Mathematical capabilities of ChatGPT. arXiv. arXiv:2301.13867v2
26. Yuan Z, Yuan H, Tan C, Wang W, Huang S. How well do large language models perform in arithmetic tasks? arXiv. 2023:arXiv:2304.02015
27. Alkaissi H, McFarlane SI (2023) Artificial Hallucinations in ChatGPT: Implications in Scientific Writing. Cureus J Med Sci 15(2).https://doi.org/10.7759/cureus.35179
28 Beutel G, Geerits E, Kielstein JT (2023) Artificial hallucination: GPT on LSD? Crit Care 27(1):148. https://doi.org/10.1186/s13054-023-04425-6