



nmbench: An Integrated Text-Editor–Based Environment for NONMEM Population Modeling and Simulation

Woojin Jung¹, Soyoung Lee^{2*}, Jung-woo Chae^{2,4*}, Hwi-yeol Yun^{2,3,4*}

¹ Graduate School of Clinical Pharmacy, CHA University

² College of Pharmacy, Chungnam National University, Daejeon, South Korea

³ Senior Health Convergence Research Center, Chungnam National University, Republic of Korea

⁴ Bio-AI Convergence Research Center, Chungnam National University, South Korea

*Those of authors contributed equally as correspondence.

BACKGROUND

1. Introduction: Model Development Environment

Pharmacometric modeling increasingly demands integrated environments that connect command-line tools with modern data science workflows. While NONMEM remains the gold standard for NLME analysis, its text-based, batch-oriented operation limits interactivity, visualization, and reproducibility. Modern IDEs such as Visual Studio Code and Positron offer a foundation for workflow modernization, and nmbench bridges NONMEM and PsN functionalities into this graphical environment for efficient, reproducible analysis.

2. NONMEM-based Workspace

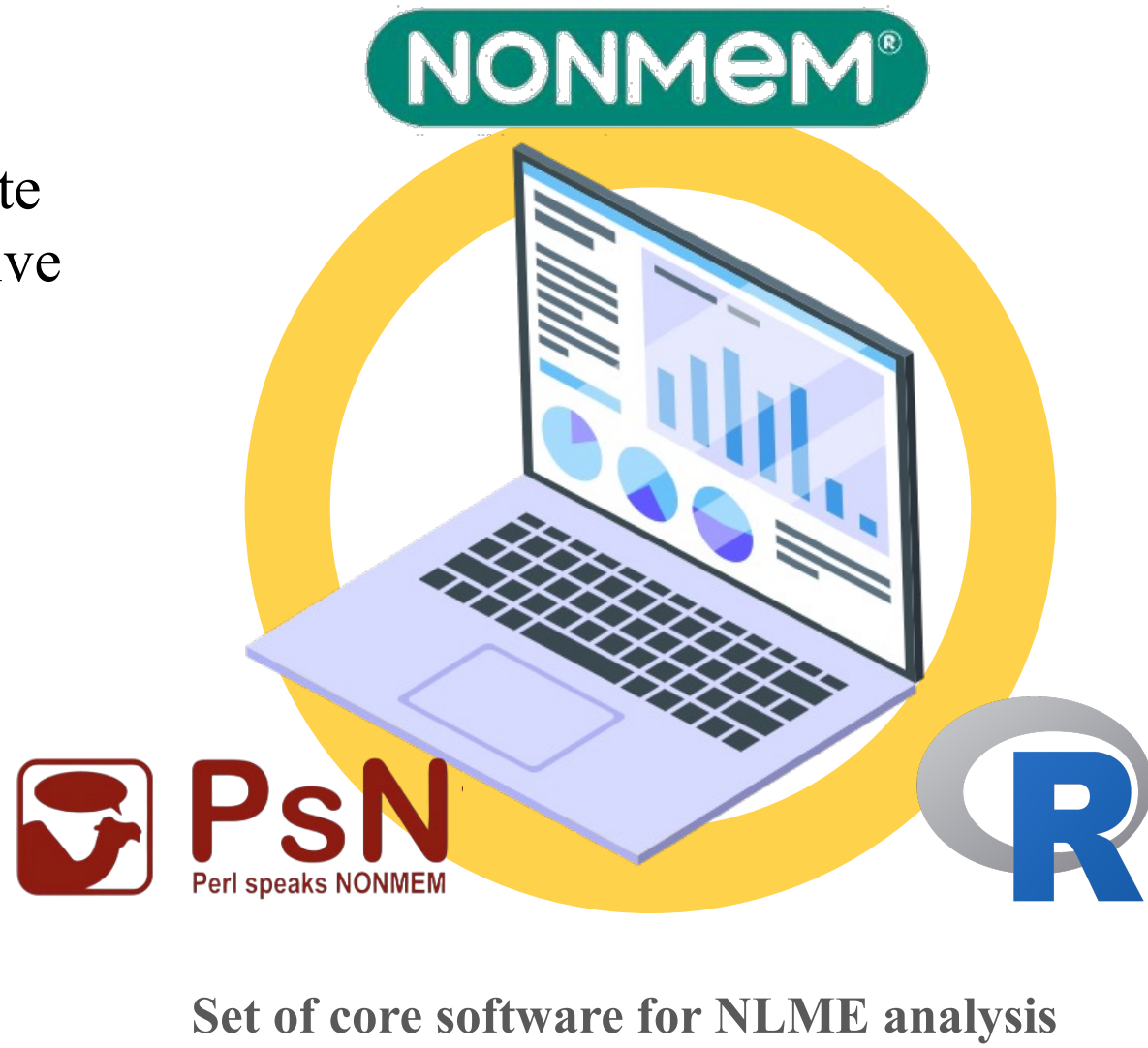
A contemporary NLME modeling environment should seamlessly integrate NONMEM, PsN, and R within a cross-platform GUI, supporting interactive visualization, reproducible and version-controlled automation, and scalability to remote systems.

3. Development Platforms: VS Code / Positron

Visual Studio Code (Microsoft) is an extensible IDE with syntax highlighting, debugging, Git integration, and a rich extension ecosystem. Positron (Posit) builds on this framework with native support for R, Python, and Quarto—together providing a flexible, cross-language platform for reproducible pharmacometric workflows.

4. Implementation: The nmbench Extension

Existing pharmacometric environments are often burdened by heavy, layered software that limits efficiency and reproducibility. Built on VS Code and Positron, nmbench unifies NONMEM, PsN, and visualization tools into a streamlined, extensible interface for model execution and diagnostics.



Set of core software for NLME analysis



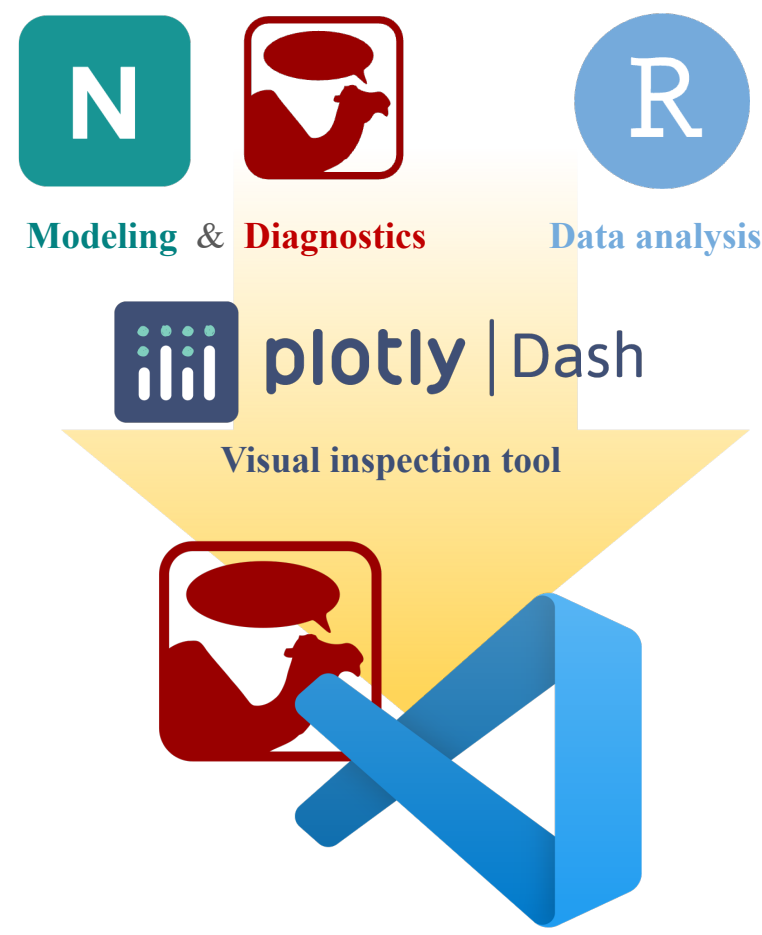
Recommendation of integrated workflow for NONMEM/R-based modeling in Code Editor (or IDE)

OBJECTIVES

This study aims to develop an integrated, user-friendly workspace that enhances the efficiency, interactivity, and reproducibility of NONMEM-based NLME modeling. We designed a lightweight NONMEM/PsN workbench that integrates seamlessly into the IDE without altering its default behavior.

Based on PsN (Perl-speaks-NONMEM) commands, it supports model execution, diagnostics, visualization, and file management—all within a single interface.

- TypeScript-based data visualization
- Tree-viewer for organized NONMEM file management
- Support for predefined R scripts for visualization or automation
- Free to use



Integration of PM workflow in VS Code

METHODS

Key Features: IDE Interface Modifications

1) Dedicated Explorer Panels

NMBENCH: BROWSER – Filters .mod / .ctl model files within a directory and supports context-menu actions such as model summary (sumo), PsN tool execution, quick file access, and R script execution (multi-select supported). NMBENCH: ESTIMATES – Parses .lst output files to instantly display parameter estimates and run status.

2) Editor Toolbar Integration

Adds quick-access buttons for Run nmfe, Run PsN tool, and Run R script directly on the editor toolbar for immediate execution.

3) Enhanced NONMEM File Management

Groups related files with shared prefixes and organizes them in a tree-view structure, simplifying iterative modeling and comparison workflows.

Provides visual cues to help users track model progress and current status.

Key Features: Results File Visualization

4) Built-in Visualization (TypeScript-based)

Automatically recognizes NONMEM-related output files and applies the optimal visualization method for each file type. Supports dynamic rendering of common outputs such as heatmaps, histograms, and time-series (concentration–time) plots.

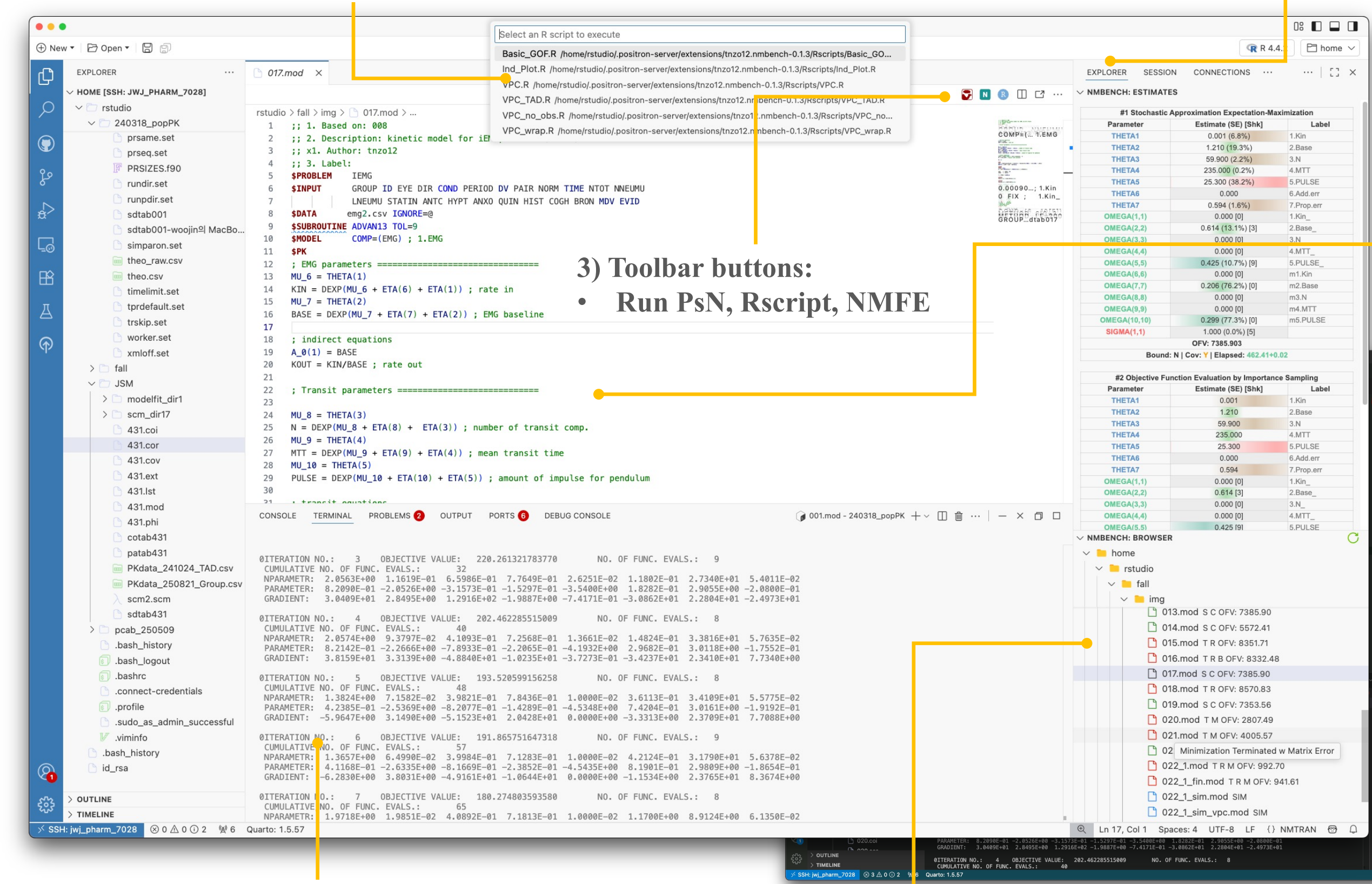
RESULTS

Interface Overview

- Drop-down menu:
 - Apply Pre-made Rscripts (Customizable)
 - View Related files (-tabs, NONMEM generated files)

2) Parameter estimates:

- Multiple \$ESTIMATION results will appear sequentially below
- Estimate change from initial is visualized (green-yellow-red)



3) Toolbar buttons:

- Run PsN, Rscript, NMFE

4) Code-editing pane:

- Text editing

5) Terminal Viwer

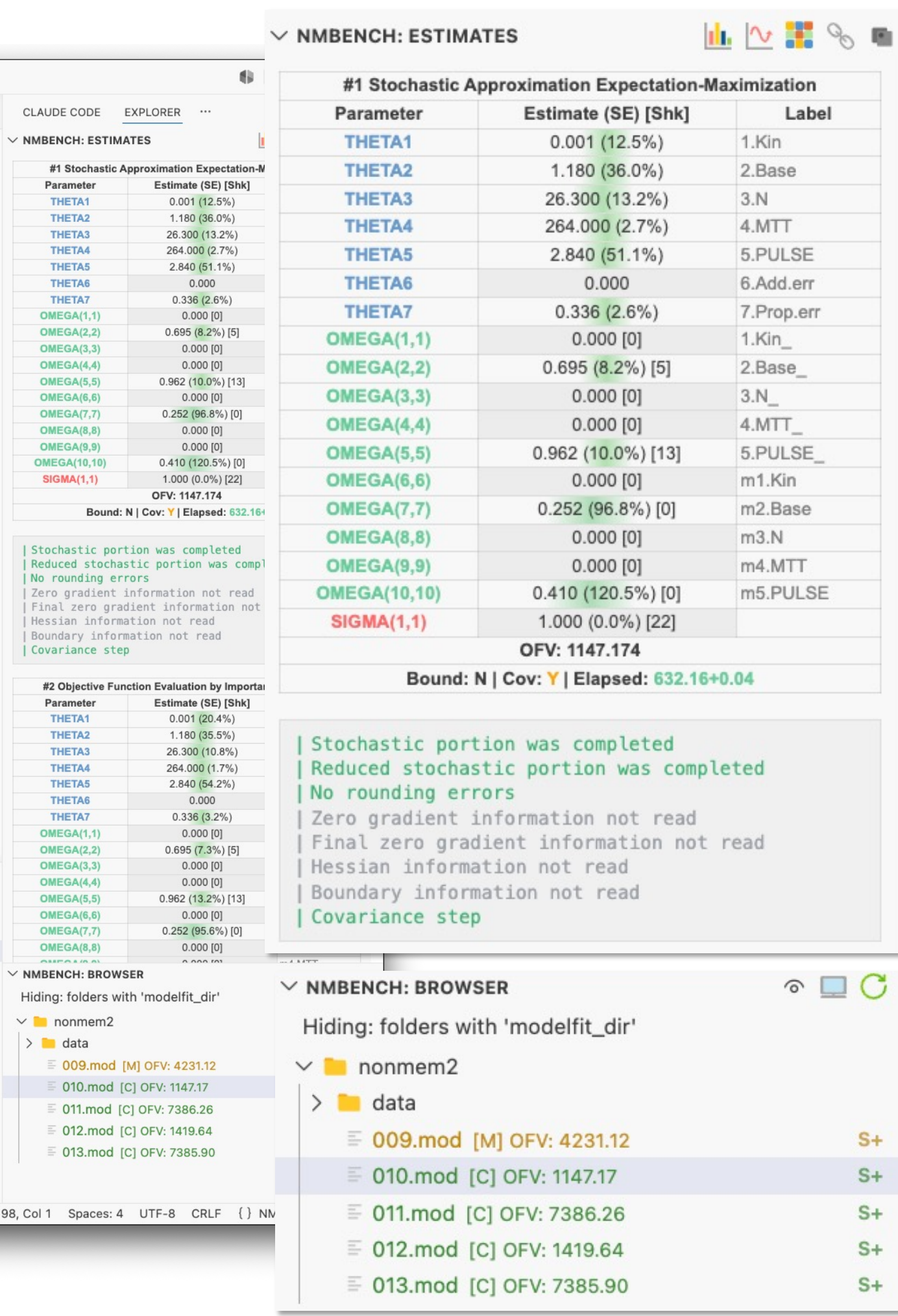
6) Tree-view:

- Manage .mod, .ctl files
- Model run summary / Run PsN commands / Show related files / Run Rscripts
- Model status (red: terminated, yellow: running, green: run successful, blue: simulation)

7) Dark theme compatible

RESULTS

Estimation Results



NMBENCH: Estimates

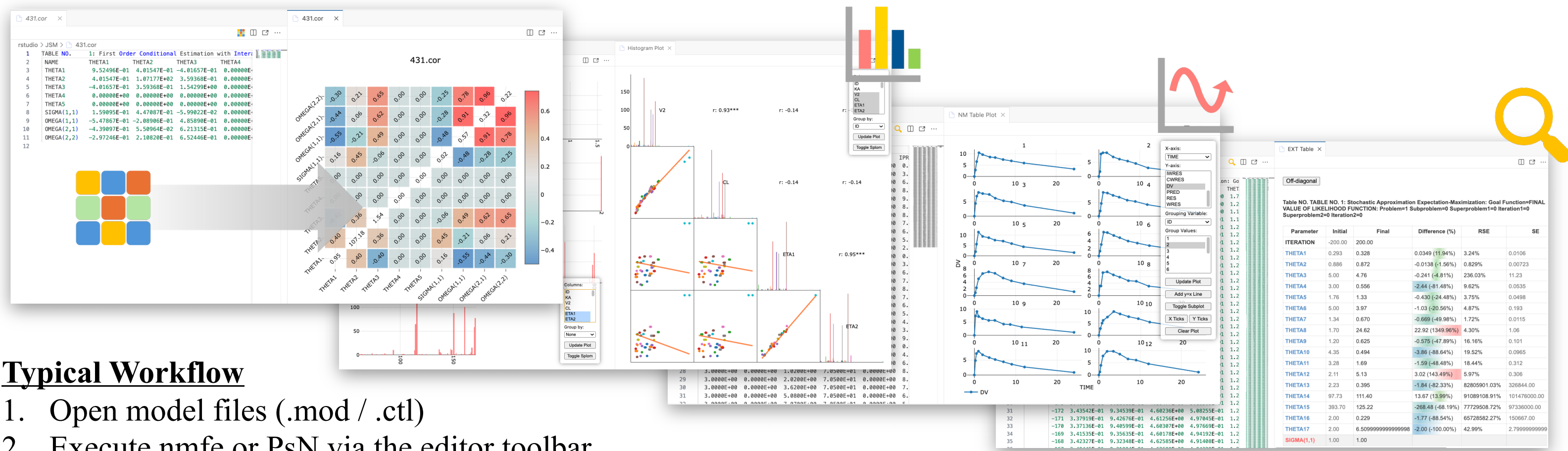
- Header
 - Linked data Inspection: Histogram, Line plot, Heatmap
 - Linked file management
 - Duplicate window to compare results
- Main body
 - Can address multiple estimations results at once: Fixed, Point estimates, RSE, Shrinkage, ETA bar...
 - Detect errors in estimation
 - Run Completion / Termination (both GD and EM methods)
 - Detect rounding errors
 - Zero gradient problem
 - Hessian reset problem
 - Boundary information
 - Covariance step

NMBENCH: Browser

- Header
 - See / Hide modelfit folders
 - Can monitor NONMEM run
 - Refresh file tree view
- Main body (File tree)
 - Can manage the model file in working directory
 - Run status overview
 - Call PsN / R functions & Make, Customize R scripts

Result File Inspection (Visualizations)

- Provides heatmaps (for matrix-type data: *.cov, *.cor, *.coi file),
 - Blue: Lower correlation / values, Red: Higher correlation / values
- Recognizes NONMEM-generated table formats (~tab / ~table named files), provides an interactive data inspector.
 - Lineplot (designed with usability for 'sdtab')
 - Histogram (for 'patab', and 'cotab')
 - SPLOM (Scatter Plot Matrix) with correlations
- .ext file viewer (similar to the Parameter estimates tab, but for model comparison)



Typical Workflow

- Open model files (.mod / .ctl)
- Execute nmfe or PsN via the editor toolbar
- Check parameter estimates (.lst) in the ESTIMATES panel
- Use visualization buttons for quick heatmaps / line plots / histograms
- Run R scripts for advanced plotting or automated report generation

Tested Environment (~System Requirements)

- NONMEM® (v7.5.1), PsN 5.3.1+, (For Rscripts) R 4.4.0+ with packages: xpose/xpose4, reshape2, dplyr, ggpubr, etc.
- Add R, PsN, and NONMEM to the system PATH
- Recommended: NMTRAN syntax extension for VS Code; for macOS, use Homebrew for R installation.
- Known limitation: "Run R script" currently supports single-model execution only.

CONCLUSION & DISCUSSION

Conclusion

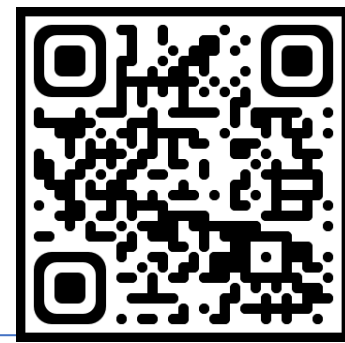
- The development of nmbench was successfully completed, providing a stable and extensible workspace for NONMEM-based NLME modeling.
- The software has been validated across multiple operating systems (Windows, macOS, and Linux) and demonstrated reliable performance in remote computing environments.
- These results confirm that the integration of NONMEM, PsN, and R visualization within Visual Studio Code can effectively enhance workflow flexibility, reproducibility, and accessibility for pharmacometric modelers.

Discussion

- The adoption of IDE-based tools such as nmbench marks a step toward modernizing the pharmacometrics workflow, bridging traditional command-line modeling with interactive, cross-platform analysis environments.
- Future development could explore tighter integration with Positron and multi-model R script execution, further improving automation and collaborative reproducibility.
- Feedback on usability, visualization design, and interoperability from the community will be essential to refine the tool and align it with evolving research and regulatory needs.

Collaboration!

- Linkedin: www.linkedin.com/in/woojin-jung-0350a2163
- GitHub: <https://github.com/tzno12/nmbench>
- VS Code Marketplace: <https://marketplace.visualstudio.com/items?itemName=tzno12.nmbench>



Acknowledgement

This study was supported by Chungnam National University, Institute of Information & Communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2022-00155857, Artificial Intelligence Convergence Innovation Human Resources Development (Chungnam National University)); the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT; No. RS-2023-00278597, RS-2022-NR070856 [Senior Health Convergence Research Center based on Life Cycle (Chungnam National University)]); the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2025-25397599); the Korea Environmental Industry & Technology Institute (KEITI) through Core Technology Development Project for Environmental Diseases Prevention and Management funded by the Korea Ministry of Environment (MOE) (RS-2021-KE001333); a grant from the Korea Machine Learning Ledger Orchestration for Drug Discovery Project (K-MELLODDY) funded by the Ministry of Health & Welfare and Ministry of Science and ICT, Republic of Korea (RS-2024-00460694); the Korea Institute of Toxicology (KIT) Research Program (No. 2710008763, KK-2401-01); the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI) grant funded by the Ministry of Health & Welfare, Republic of Korea (grant number: RS-2024-00336984, RS-2025-02306055), support from the Ministry of Trade, Industry, and Energy (MOTIE), Korea, under the "Infrastructure program for industrial innovation" supervised by the Korea Institute for Advancement of Technology (KIAT) (RS-2024-00434342); and a grant (No. 26212MFDSD008) from the Ministry of Food and Drug Safety (MFDS), Republic of Korea.