

A VALIDATION FRAMEWORK FOR AI-BASED SCIENTIFIC TOOLS IN MONOLIXSUITE

Authors: Stephanie Kollmann (1), Frano Mihaljevic (1), Monika Twarogowska (1)
Institutions: (1) Simulations Plus, Product & Technology division, France



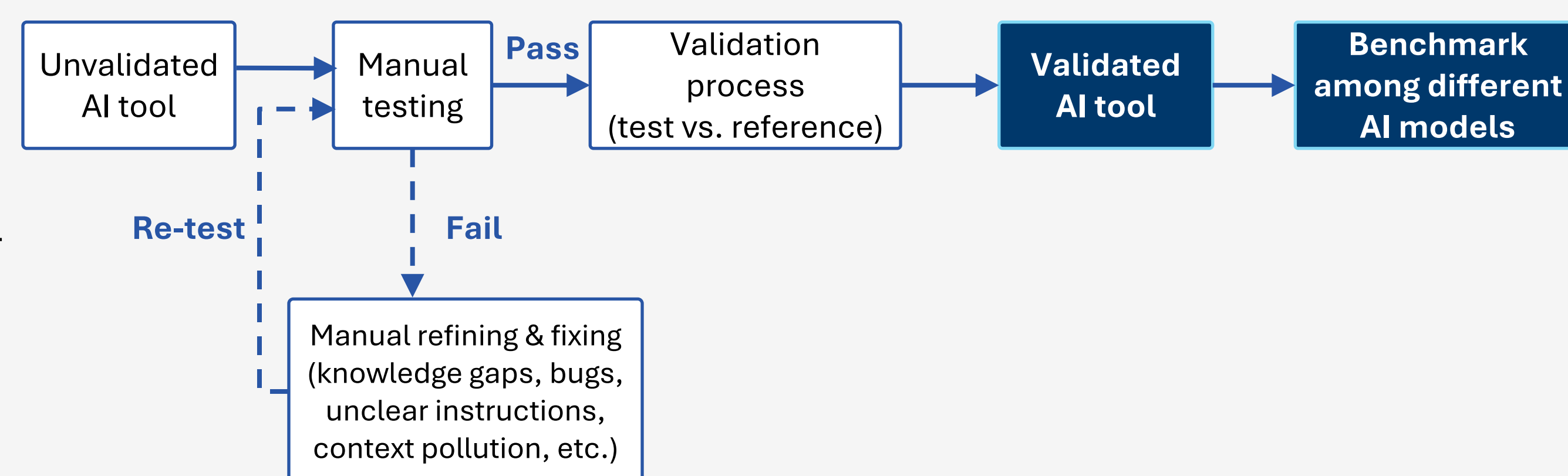
SimulationsPlus

INTRODUCTION

AI tools are increasingly common in pharmacometrics, with the potential to support routine tasks and make complex workflows more efficient [1]. However, **systematic validation of their performance in real-world workflows remains limited** [2]. This is especially important for technical AI tools, where outputs may look correct but fail in practice.

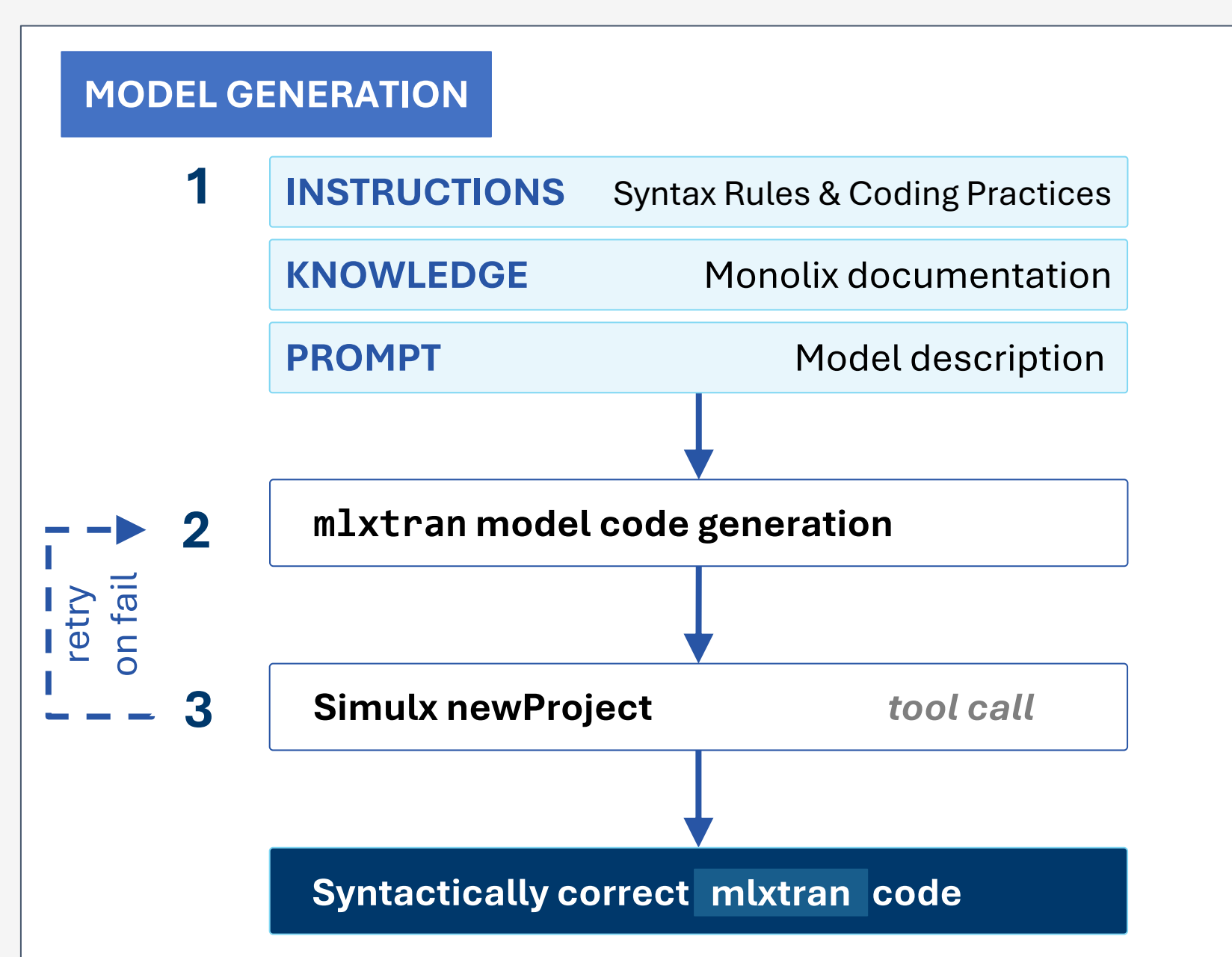
A **reusable validation procedure** was developed using reference examples and automated checks.

Two AI tools were **systematically validated and benchmarked** to assess their **reliability and reproducibility** in MonolixSuite.



mlxtranModelWriter Tool

The **mlxtranModelWriter** tool can generate an **mlxtran** model from a **natural-language** description or **NONMEM** code or **edit an existing mlxtran model**. The generated or modified model is checked for syntax errors and, if errors are detected, revised iteratively until valid code is obtained.



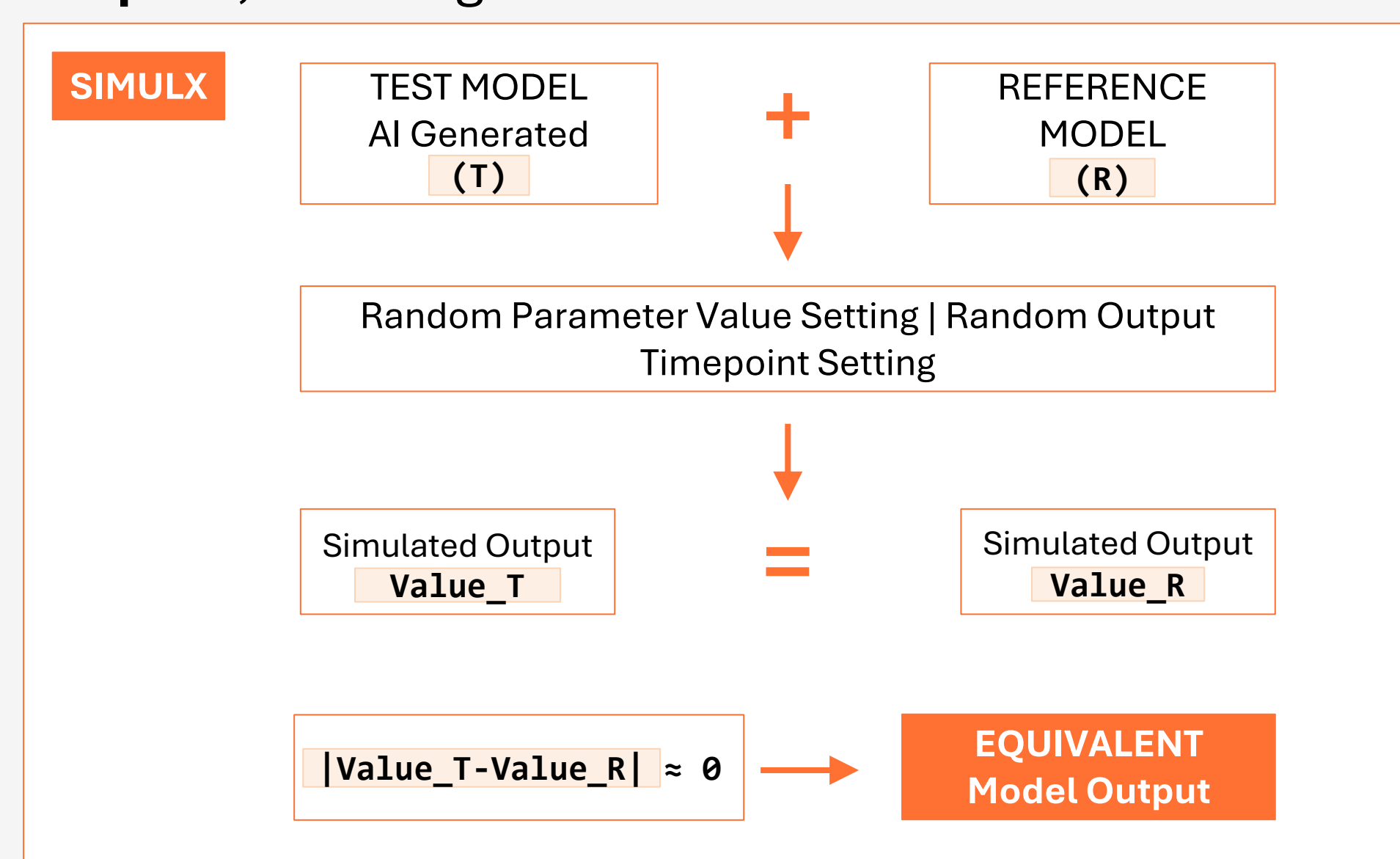
Validation process:

Validation dataset:

20 model examples with a **natural-language description** and a **correct reference mlxtran model**.

Validation challenge:

Different model implementations can be correct → Validation compares **simulated outputs**, not the generated model.

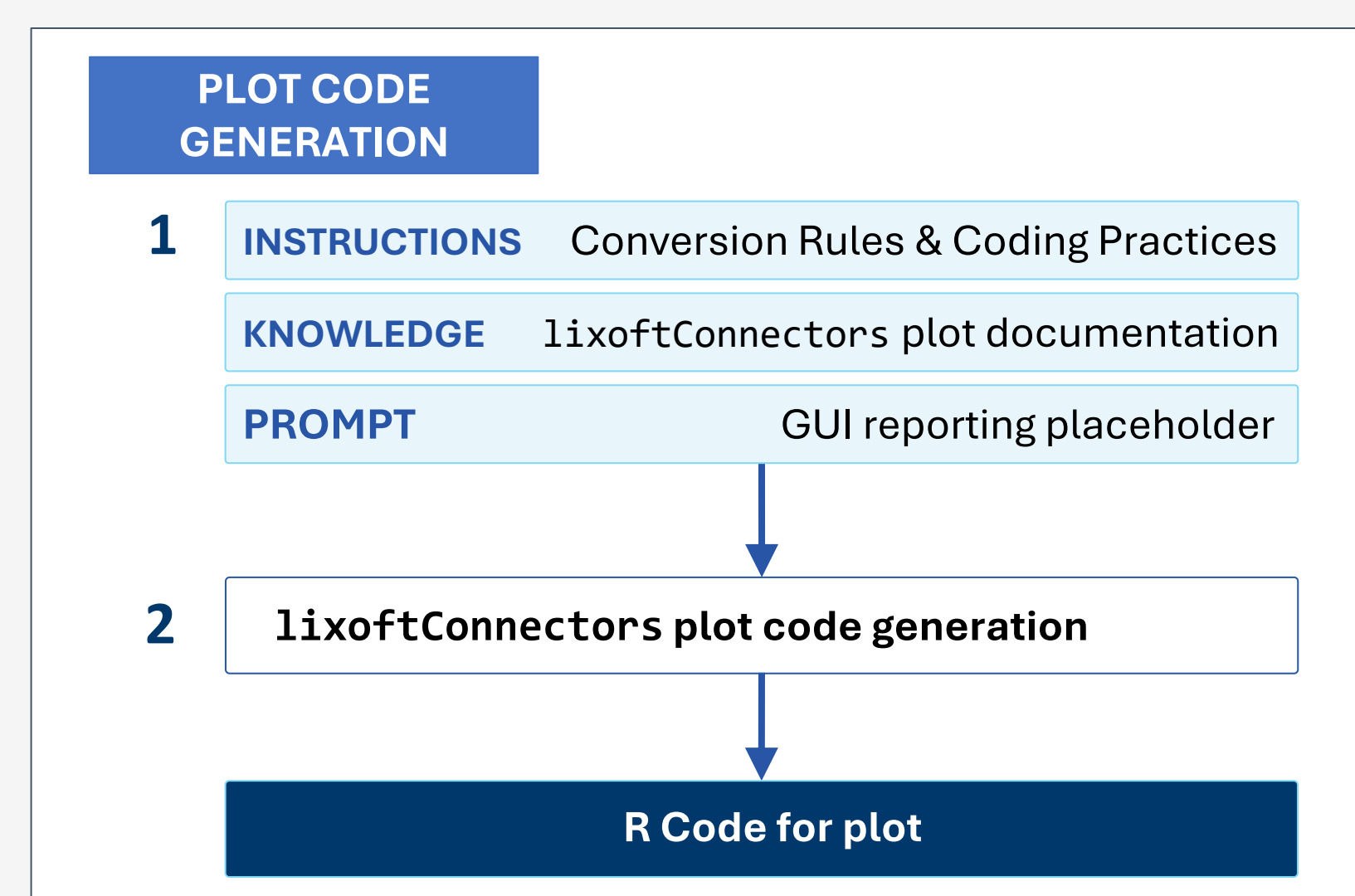


Benchmark:

6 AI model-reasoning configurations × 20 examples × 10 replicates
→ **1200 generated models**.

guiPlotConverter Tool

The **guiPlotConverter** tool converts **GUI-generated plot placeholder code** into **lixoftConnectors plot R code**, the **R API for MonolixSuite**. It uses predefined **conversion rules and coding practices** together with the **lixoftConnectors plot documentation** to generate an equivalent plot implementation.



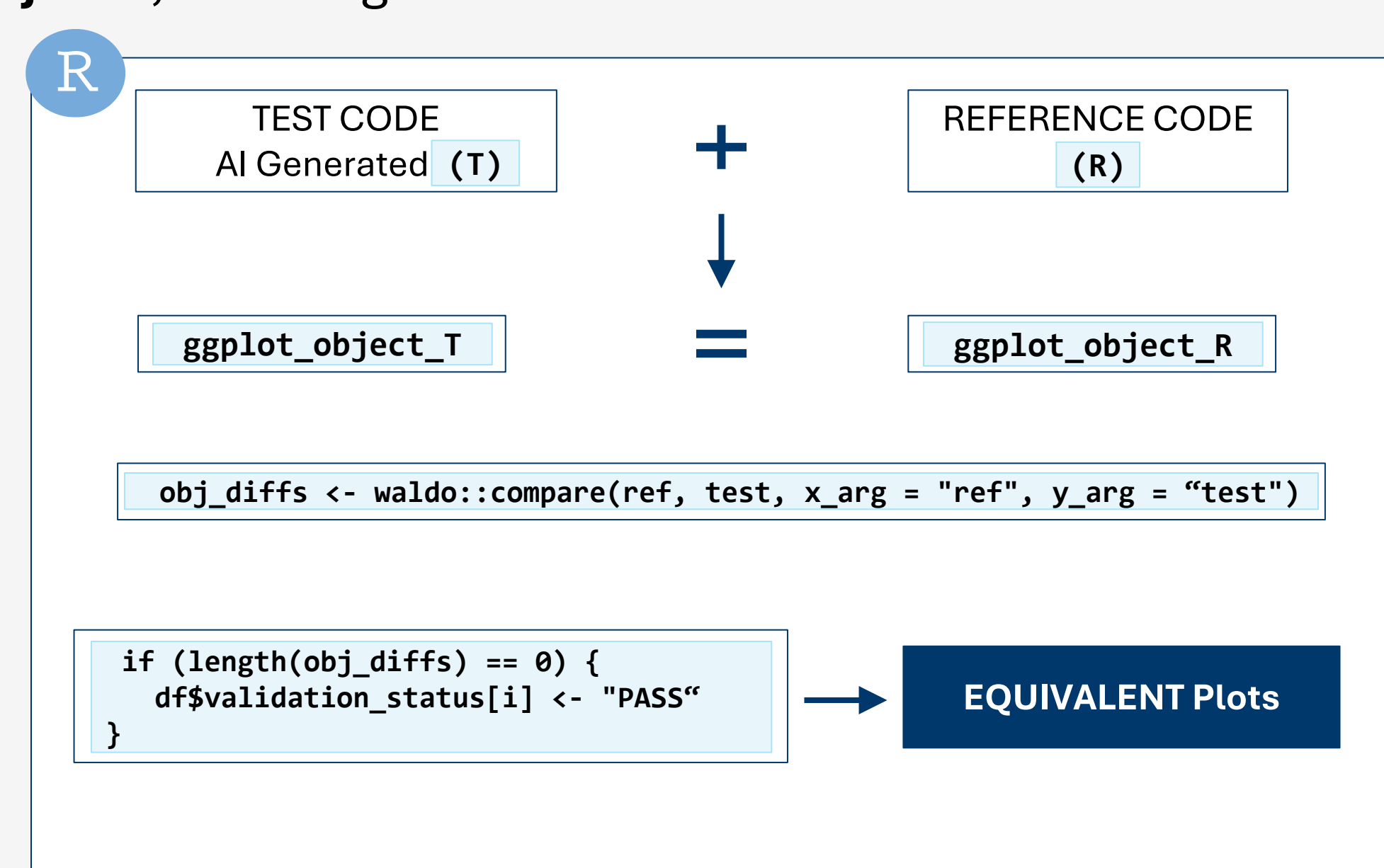
Validation process:

Validation dataset:

100 plot examples with a **placeholder from plot customized in GUI** and corresponding **reference lixoftConnectors plot R code**.

Validation challenge:

Different plot code implementations can be correct → Validation compares **ggplot objects**, not the generated code.

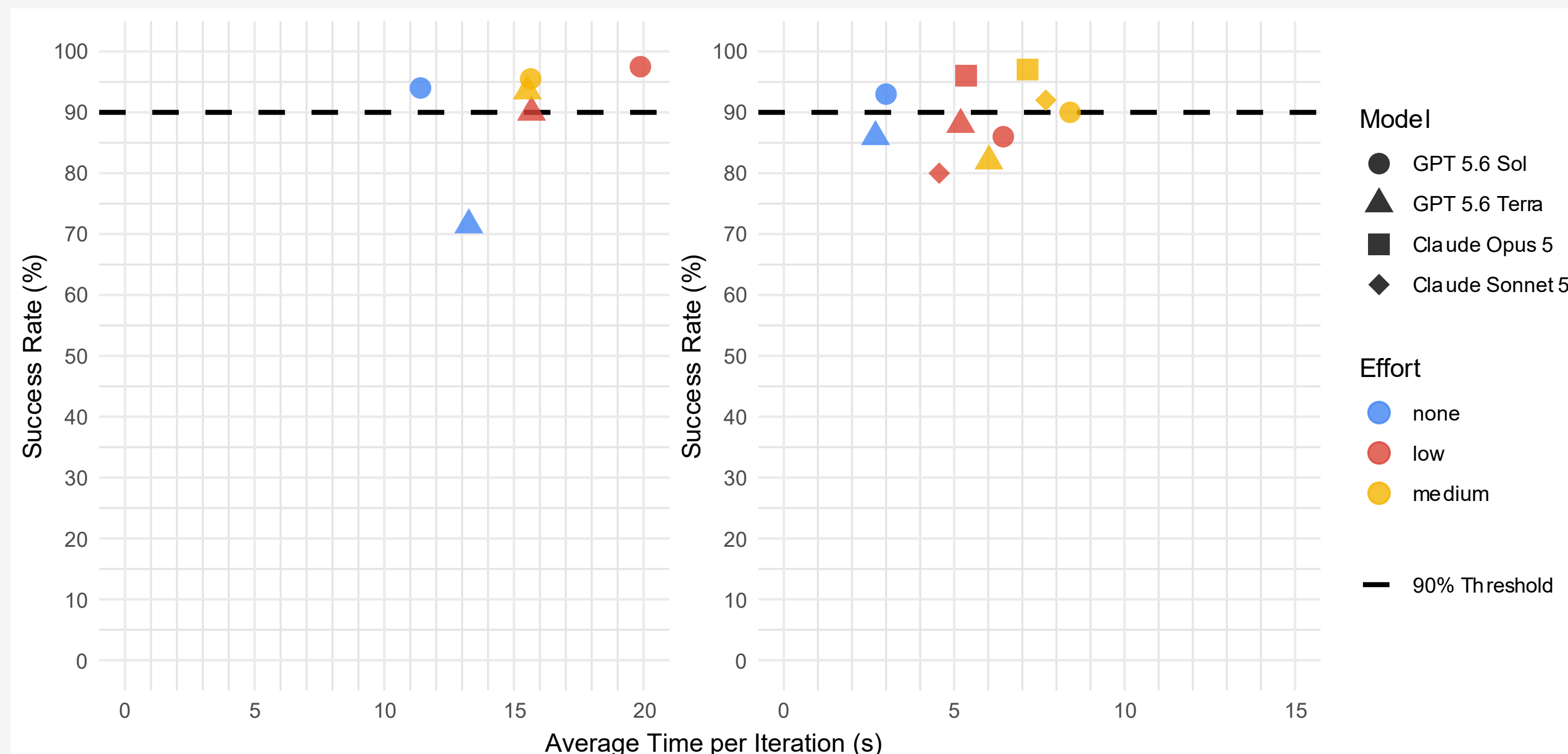


Benchmark:

10 AI model-reasoning configurations × 100 examples → **1000 generated plots**.

RESULTS

GPT-5.6 Sol and Terra were evaluated with **none, low, and medium reasoning effort**; Claude Opus 5 and Sonnet 5 with **low and medium effort**. For **mlxtranModelWriter**, success rates were **averaged across replicates**. **≥90%** was set as the acceptance threshold.



mlxtranModelWriter

guiPlotConverter

- Best accuracy:**
Claude Opus 5 - medium (97%) on converter; GPT 5.6 Sol - low (97.5%) on writer.
- Worst accuracy:**
Claude Sonnet 5 - low (80%) on converter; GPT 5.6 Terra - none (71.5%) on writer.
- Speed:**
fastest is GPT Terra/Sol - none (2.7 s converter, 11.4 s writer); slowest is GPT Sol - medium converter (8.4 s) and Sol - low writer (19.9 s).
- Best compromise:**
GPT 5.6 Sol - none in both ≥ 93% accuracy at the fastest usable speed

CONCLUSION

- Validation and benchmarking make tool performance measurable and reproducible.
- Process should be rerun after changes to the AI tool (bug fixes, knowledge gaps, etc.) or to test new LLM models.
- They help identify the best model and configuration for each task.
- Model choice can balance accuracy, runtime, and cost.

References:

- [1] Tosca EM, et al. Pharmacometrics in the age of large language models. *Pharmaceutics*. 2025.
- [2] Wang Y, et al. Predictive AI in clinical pharmacology: a call to action. *CPT Pharmacometrics Syst Pharmacol*. 2025.

Contact information:

stephanie.kollmann@simulations-plus.com
frano.mihaljevic@simulations-plus.com



www.simulations-plus.com