

Enhancing Model Interoperability and Workflow Integration in Clinical Trial Simulations

Jean Ponchon^{1*}, Elodie Prevot¹, Matthieu Coudron¹, Jeremy Villard¹, Nicolas Duteil¹, Riad Kahoul¹, **Jean-Baptiste Gourlet¹, Frederic Cogny¹**
* Corresponding authors, ¹ Nova In Silico, Lyon, France

Import once, simulate anywhere in hours

Jinkō ensures **consistent results** across R, MATLAB, and its native interface, accelerating interoperability with any simulation workflows.

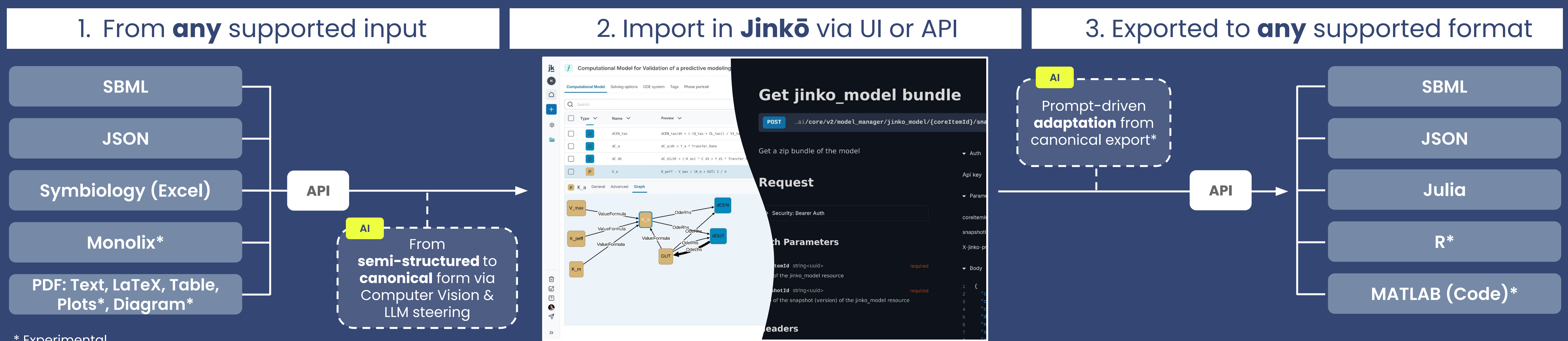


Fig. 1: Interoperability through an API-first platform

Diverse inputs are ingested and normalized into the Jinkō API-driven platform, then exported to multiple formats (SBML, Julia, MATLAB, R) with optional prompt-based adaptation.

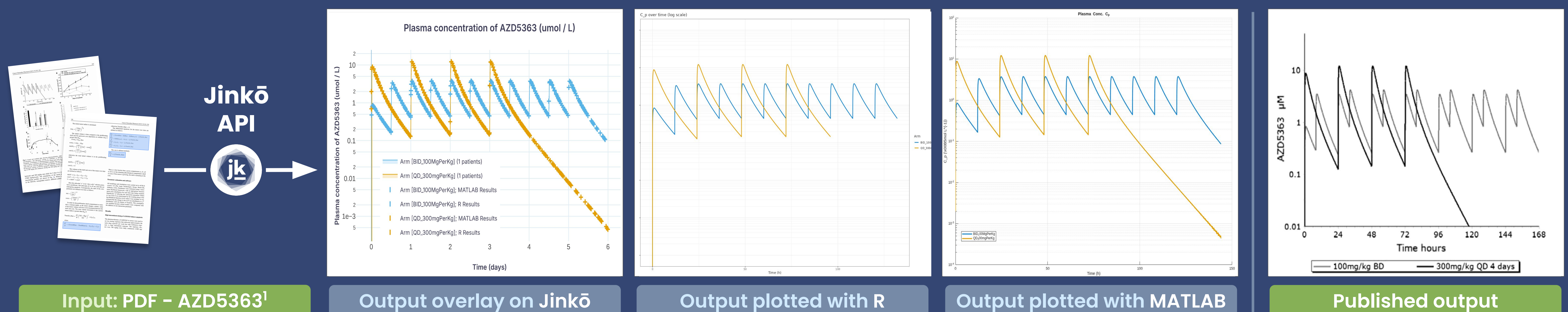


Fig. 2: Cross-tool reproducibility case study with Jinkō: From a publication to R and MATLAB scripts

A published model of AZD5363¹ was imported as PDF. Reconstructed automatically to a canonical model (< 3 min), reviewed and corrected by a modeler (< 2 hours). Simulated in Jinkō, and then exported as standalone scripts to R and MATLAB (< 30 minutes). All executions reproduced the published output, with overlay plots back to Jinkō confirming identical results across tools.

BACKGROUND

Interoperability across modeling tools remains a barrier to reproducibility. Standards such as PharmML and SBML help^{2,3,4} but adoption is uneven; models still require manual re-implementation across environments (e.g., Monolix, SimBiology, MATLAB/R/Julia), introducing errors, delays, and limiting reuse⁵.

Jinkō is a **language-agnostic, API-first** platform with **polyglot import/export** that integrates with existing modeling and simulation workflows. Building on this foundation, we evaluate **LLM-assisted transpilation** to close remaining interoperability gaps.

METHODS

Data ingestion and normalization. The platform ingests structured sources (SBML, Monolix, SimBiology exports, serialized JSON/YAML) and semi-structured sources (scientific publications in PDF/LaTeX). All inputs are normalized into a canonical model representation exposed via the Jinkō API, covering equations, parameters, units, virtual populations, and trial designs for programmatic editing and automation.

Large language models (LLMs) are applied at three levels:

- **Parser development:** accelerate implementation, maintenance, and validation.
- **Semi-structured import:** extract equations, parameters, and units from publications.
- **Last-mile transformation:** adapt canonical models into user-preferred code structures in Julia, MATLAB, or R.

Validation & benchmarking. Robustness of the platform is assessed with 13,701 automated tests (unit, golden, end-to-end) and external evaluations: SBML Features Test Suite (1,545/1,821 features passed; 84.8%) and Curated BioModels import (80% success, n=1,067).

RESULTS

Case study: A published model was imported directly from **PDF**, reconstructed in Jinkō, and executed. The same canonical model was then exported to **MATLAB** and **R** and executed natively.

- **Overlay validation:** Outputs from Jinkō, MATLAB, and R were **identical** within solver precision and matched the published results.
- **Efficiency:** End-to-end semi-automated workflow from PDF import to reproducible execution across three environments completed in **<3 hours** (vs. about **1–2 days** for manual transcription).
- **Adaptability:** Exported code can be modified for new analyses, visualizations, or datasets in **minutes**, replacing hours of error-prone adjustments.

CONCLUSION

Interoperability in modeling and simulation can be **accelerated and partially automated** through:

- A **language-agnostic canonical representation** exposed via an API, supporting SBML and prominent programming languages.
- A **rigorously validated platform** (13k+ tests plus external benchmarks) to ensure correctness and reproducibility.
- **LLM-enabled workflows** for parser development, semi-structured import, and user-customized exports.
- **API-first integration**, enabling these pipelines to embed cleanly into broader modeling and simulation ecosystems.

Poster and generated models available at:
<https://github.com/novainsilico/2025-poster-model-interoperability>



REFERENCES

- [1] Yates, James W. T., Philippa Dudley, Jane Cheng, Celina D'Cruz, and Barry R. Davies. "Validation of a Predictive Modeling Approach to Demonstrate the Relative Efficacy of Three Different Schedules of the AKT Inhibitor AZD5363." *Cancer Chemotherapy and Pharmacology* 76, no. 2 (June 20, 2015): 343–56. <https://doi.org/10.1007/s00280-015-2795-7>.
- [2] Swat, M J et al. "Pharmacometrics Markup Language (PharmML): Opening New Perspectives for Model Exchange in Drug Development." *CPT: pharmacometrics & systems pharmacology* vol. 4,6 (2015): 316–9. doi:10.1002/psp4.57
- [3] Bizzotto, R et al. "PharmML in Action: an Interoperable Language for Modeling and Simulation." *CPT: pharmacometrics & systems pharmacology* vol. 6,10 (2017): 651–665. doi:10.1002/psp4.12213
- [4] Terranova, Nadia et al. "The Standard Output: A Tool-Agnostic Modeling Storage Format." *CPT: pharmacometrics & systems pharmacology* vol. 7,9 (2018): 543–546. doi:10.1002/psp4.12339
- [5] Kirouac, Daniel C et al. "Reproducibility of Quantitative Systems Pharmacology Models: Current Challenges and Future Opportunities." *CPT: pharmacometrics & systems pharmacology* vol. 8,4 (2019): 205–210. doi:10.1002/psp4.12390

