A Demonstration of an LLM-based Multi-agent System for Drug Discovery

Demonstration Track

Lakshidaa Saigiridharan
MolecularAI, Discovery Sciences, AstraZeneca R&D
Gothenburg, Sweden
lakshidaa.saigiridharan@astrazeneca.com

Helen Lai
MolecularAI, Discovery Sciences, AstraZeneca R&D
Cambridge, United Kingdom
helen.lai1@astrazeneca.com

Kinga Jenei
MolecularAI, Discovery Sciences, AstraZeneca R&D
Gothenburg, Sweden
kinga.jenei@astrazeneca.com

Jiazhen He
MolecularAI, Discovery Sciences, AstraZeneca R&D
Gothenburg, Sweden
jiazhen.he@astrazeneca.com

Samuel Genheden
MolecularAI, Discovery Sciences, AstraZeneca R&D
Gothenburg, Sweden
samuel.genheden@astrazeneca.com

ABSTRACT

In drug discovery, AI tools are used for many tasks such as generating compound ideas, predicting their properties, and planning their synthesis. We demonstrate an agentic system based on a large language model that is capable of orchestrating such tools. We discuss how this system was implemented for use in everyday tasks by AstraZeneca scientists. We put particular emphasis on evaluating such an agentic system to ensure robustness and reproducibility, and to this end we release an open-source version of the code. A video demonstrating the system is available here: https://doi.org/10.5281/zenodo.18195147

KEYWORDS

drug discovery; large language models; agents; agentic AI

ACM Reference Format:
Lakshidaa Saigiridharan, Helen Lai, Kinga Jenei, Jiazhen He, and Samuel Genheden. 2026. A Demonstration of an LLM-based Multi-agent System for Drug Discovery: Demonstration Track. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/KAPY7208

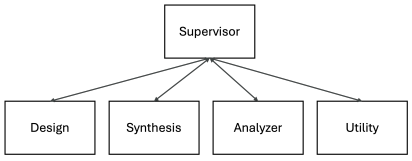
1 INTRODUCTION

In drug discovery, advancing a molecule toward a clinical candidate involves an iterative design–make–test–analyze (DMTA) cycle, in which scientists design new compounds, synthesize them, evaluate them experimentally, and use the results to guide subsequent design rounds [16]. Computational tools have long supported this process by assisting with compound generation, evaluation, and synthesis planning [4, 20]. Over the past decade, advances in deep learning have shifted these approaches from rule-based systems to data-driven models [14].

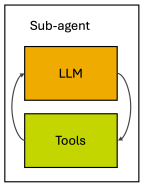
Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/KAPY7208

With the technological advancements in large language models (LLMs) in just the last few years [17, 19], it is of interest to investigate if we stand before another paradigm shift. Several LLM-based agents have been proposed for drug discovery and related chemistry tasks [2, 3, 5, 13, 15]. In these systems, the LLM is orchestrating different computational tools that gives the LLM a more grounded basis for decision making.

In this demonstration, we present an LLM-based multi-agent system for drug discovery developed at AstraZeneca [6]. We focus on introducing an evaluation framework to assess the robustness and reliability of the agentic system. To support transparency and reproducibility, we release an open-source version of the codebase that reflects the structural complexity of our internal platform. The presented system supports routine drug discovery tasks, including analog generation, compound scoring, and synthesis planning.



(a) Overview of the multi-agent architecture



(b) Sub-agent consists of an LLM and tools that exchange information iteratively

Figure 1: Overview of the agentic system

2 AGENT OVERVIEW

We implemented the system as a multi-agent architecture in which a supervisor coordinates several specialized sub-agents (Fig. 1a).

A user's natural language prompt is processed by the supervisor, which selects an appropriate sub-agent and generates a task description based on the request. After receiving the sub-agent's output, the supervisor evaluates the result and iteratively routes subsequent tasks until a final response is produced. Each sub-agent uses an LLM to translate instructions into a sequence of calls to external tools, such as Python scripts or production-grade AI platforms, and returns the processed results to the supervisor (Fig. 1b). The architecture evolved from an earlier single-agent design to improve scalability, robustness, and token efficiency [6].

The core system consists of the following four sub-agents:

- Design - an agent that can generate novel compounds and score compounds
- Synthesis - an agent that can assist in the planning of the chemical synthesis of compounds
- Analyze - an agent that can analyze an output file in CSV format in order to compute statistics or make plots
- Utility - an agent that assists in miscellaneous tasks such as converting common names of compounds to chemical structures

The *Design* agent uses different running modes of the REINVENT 4 platform [12] to either generate novel compounds or score them. Note that many scoring functions are proprietary or require a software license, therefore, the open-source code only supports some basic scoring functions. The *Synthesis* agent uses the AiZynthFinder software [18] to perform retrosynthesis in order to predict a synthesis plan. It also uses the Precedent Finder algorithm [1] to find close analogs of chemical reactions in database(s) of historical reaction data. The *Analyze* agent is provided as part of the LangChain ecosystem [9] and uses an LLM to generate python code that can be executed in a local python executable. The python code is used to do various manipulations of a Pandas dataframe. The *Utility* agent validates chemical structures in SMILES (Simplified Molecular Input Line Entry System) format [21] and can query PubChem [8] to retrieve SMILES strings from common molecule names.

The agent code was implemented in Python using the LangChain framework [9]. The tools were provided to the agent as MCP (model context protocol) servers. We used GPT-4o [7] from OpenAI as the LLM.

It is important to note that the capabilities described above represent only a subset of those available in the production system, reflecting restrictions on internal API access. Discussions with a core user group indicated that these functionalities alone would be insufficient to support routine, day-to-day use, which motivated the development of additional capabilities, such as matched molecular pair analysis, as well as planned integrations with multiple internal databases. Nevertheless, the functionalities presented here constitute the foundational core of the system, upon which a broader set of capabilities is built. This extensibility is facilitated by the modular multi-agent architecture, which enables additional sub-agents to be readily developed to further extend the system's functionality.

3 BENCHMARKING

LLM-based systems such as our multi-agent system are inherently more challenging to test and evaluate than other software due to the stochastic nature of LLMs. We believe that properly benchmarking these systems is essential for ensuring robustness and instilling trust in end-users. As part of this evaluation, we created a suite of test questions that represent typical user prompts. A few example questions are provided in Table 1

Table 1: Example questions used to benchmark the agent

Question	*Sub-agent called*
What is the SMILES of Ibuprofen?	Utility
Can you generate molecules similar to [SMILES string]	Design
How many compounds have MW between 500 and 600, HBD < 3 in smiles.csv?	Analyzer
Can you generate molecules similar to Gleevec but with lower logD?	Utility → Design
Can you generate molecules similar to Gleevec with lower HBA, higher MW, logD between 1 and 3, number of rotatable bonds <= 9?	Utility → Design → Analyzer

Furthermore, we implemented optional tool mock outputs that can be used during testing. This keeps the execution of the tools completely deterministic and also increases execution speed for testing. With this setup, we used LangFuse [10] to track all the executions of the agent and we then analyze those traces. We evaluate the multi-agent system by comparing its selected tool sequence against predefined ground-truth sequences, assessing its ability to coordinate task execution correctly. We are in the process of implementing an LLM-as-a-judge system [11] for evaluating the final answer of the agent.

It is important to clarify that our benchmarking is designed to assess how effectively the agent completes the tasks it is designed to perform, rather than to compare our architecture with other published systems. This distinction is important because agentic systems are tightly coupled to their specific toolsets, prompts, and intended workflows. These factors fundamentally shape agent behavior, making architectural performance difficult to evaluate in isolation. Consequently, the notion of a universally "state-of-the-art" architecture in this domain is not well defined.

4 CONCLUSIONS

We have demonstrated the implementation of an LLM-based multi-agent system for drug discovery that is capable of designing novel compounds and suggesting synthesis plans. We developed it for internal use by AstraZeneca scientists but also provide an open-source version of the software to further open science and benchmark of these novel AI capabilities. In particular, the benchmarking of these agentic systems is essential for building trust with end-users and we have outlined our benchmarking strategy. We envisage further development of these agentic systems before they become transformative for drug discovery.

CODE AVAILABILITY

The public code will be released on GitHub prior to the demonstration and will be available at: https://github.com/MolecularAI/langdmta-lab.

ACKNOWLEDGMENTS

We acknowledge developments on an earlier version of our system by Gian Marco Ghiandoni and Jon Paul Janet. We also acknowledge development of the graphical user interface, not described herein, by Umur Gokalp and Ajsa Nukovic.

REFERENCES

[1] Christoph A Bauer, Thierry Kogej, Samuel Genheden, and Per-Ola Norrby. 2025. Precedent Finder: Locating Pareto-Optimal Reactions. *Journal of chemical information and modeling* (2025).
[2] Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous chemical research with large language models. *Nature* 624 (2023), 570 – 578.
[3] Andrés M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. 2023. Augmenting large language models with chemistry tools. *Nature Machine Intelligence* 6 (2023), 525 – 535.
[4] Carmen Cerchia and Antonio Lavecchia. 2023. New avenues in artificial-intelligence-assisted drug discovery. *Drug discovery today* (2023), 103516.
[5] Yao Fehlis, Charles Crain, Aidan Jensen, Michael Watson, James Juhasz, Paul Mandel, Betty Liu, Shawn Mahon, Daren Wilson, Nick Lynch-Jonely, Ben Leedom, and David Fuller. 2025. Accelerating Drug Discovery Through Agentic AI: A Multi-Agent Approach to Laboratory Automation in the DMTA Cycle. *ArXiv* abs/2507.09023 (2025).
[6] Jiazhen He, Helen Lai, Lakshidaa Saigiridharan, Gian Marco Ghiandoni, Kinga Jenei, Umur Gokalp, Ajsa Nukovic, Ola Engkvist, Jon Paul Janet, and Samuel Genheden. 2026. Democratising real-world drug discovery through agentic AI. *Drug discovery today* (2026), 104605.
[7] OpenAI Aaron Hurst and Adam Lerer et al. 2024. GPT-4o System Card. *ArXiv* abs/2410.21276 (2024).
[8] Sunghwan Kim, Paul A. Thiessen, Evan E. Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A. Shoemaker, Jiyao Wang, Bo Yu, Jian Zhang, and Stephen H. Bryant. 2015. PubChem Substance and Compound databases. *Nucleic Acids Research* 44 (2015), D1202 – D1213.
[9] langchain 2025. LangChain. Retrieved January 01, 2026 from http://www.langchain.com
[10] langfuse 2025. LangFuse. Retrieved January 01, 2026 from http://www.langfuse.com
[11] Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods. *ArXiv* abs/2412.05579 (2024).
[12] Hannes H. Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H. Mervin, and Ola Engkvist. 2024. Reinvent 4: Modern AI–driven generative molecule design. *Journal of Cheminformatics* 16 (2024).
[13] Andrew D. McNaughton, Gautham Ramalaxmi, Agustin Kruel, Carter R. Knutson, Rohith Anand Varikoti, and Neeraj Kumar. 2024. CACTUS: Chemistry Agent Connecting Tool Usage to Science. *ACS Omega* 9 (2024), 46563 – 46573.
[14] Lewis H. Mervin, Samuel Genheden, and Ola Engkvist. 2022. AI for drug design from explicit rules to deep learning. *Artificial Intelligence in the Life Sciences* (2022).
[15] Qihua Pan, Dong Xu, Jenna Xinyi Yao, Lijia Ma, Zexuan Zhu, and Junkai Ji. 2025. FROGENT: An End-to-End Full-process Drug Design Agent. *ArXiv* abs/2508.10760 (2025).
[16] Alleyn T. Plowright, Craig Johnstone, Jan Kihlberg, Jonas A Pettersson, Graeme R. Robb, and Richard A. Thompson. 2012. Hypothesis driven drug design: improving quality and effectiveness of the design-make-test-analyse cycle. *Drug discovery today* 17 1-2 (2012), 56–62.
[17] Mayk Caldas Ramos, Christopher J. Collison, and Andrew D. White. 2024. A review of large language models and autonomous agents in chemistry. *Chemical Science* 16 (2024), 2514 – 2572.
[18] Lakshidaa Saigiridharan, Alan Kai Hassen, Helen Lai, Paula Torren-Peraire, Ola Engkvist, and Samuel Genheden. 2024. AiZynthFinder 4.0: developments based on learnings from 3 years of industrial application. *Journal of Cheminformatics* 16 (2024).
[19] Bosheng Song, Xiaowen Li, Xiuxiu Chao, Li Wang, Yiping Liu, Zhen Xia, Dongsheng Cao, and Xiangzheng Fu. 2025. Advancements in Large Language Models (LLMs): Empowering Drug Discovery. *WIREs Computational Molecular Science* (2025).
[20] W. Patrick Walters and Regina Barzilay. 2021. Critical assessment of AI in drug discovery. *Expert Opinion on Drug Discovery* 16 (2021), 937 – 947.
[21] David Weininger. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* 28 (1988), 31–36.