From Single to Multi-Agent Reasoning: Advancing GeneGPT for Genomics QA

Kimia Abedini[1], Farzad Shami[2], and Gianmaria Silvello[1]

[1] University of Padua, Italy
[2] Aalto University, Finland

Abstract. Comprehending genomic information is essential for biomedical research, yet extracting data from complex distributed databases remains challenging. Large language models (LLMs) offer potential for genomic Question Answering (QA) but face limitations due to restricted access to domain-specific databases. GeneGPT is the current state-of-the-art system that enhances LLMs by utilizing specialized API calls, though it is constrained by rigid API dependencies and limited adaptability. We replicate GeneGPT and propose `GenomAgent`, a multi-agent framework that efficiently coordinates specialized agents for complex genomics queries. Evaluated on nine tasks from the GeneTuring benchmark, `GenomAgent` outperforms GeneGPT by 12% on average, and its flexible architecture extends beyond genomics to various scientific domains needing expert knowledge extraction.

Keywords: Question Answering · Genomic QA · Multi-Agent Systems

https://kimia-abedini.github.io/Genom-Agent/

1 Introduction

Large Language Models (LLMs) have shown remarkable potential in QA tasks and have recently gained traction in genomic QA applications [12,1]. A notable and widely cited example is GeneGPT [10], which currently represents the state-of-the-art for genomic QA tasks by successfully augmenting LLMs with external domain-specific APIs through in-context learning [3] and tool integration [19]. GeneGPT operates as a single-agent architecture[15] where an LLM is guided through carefully constructed prompts containing API documentation and examples, with inference managed sequentially through a single forward loop of API calls and result processing. Despite its effectiveness in achieving high accuracy on genomic benchmarks, GeneGPT's architecture exhibits several limiting characteristics that constrain its scalability and adaptability. The system's rigid dependency on specific API formats makes it fragile when interfacing with evolving tools, while its reliance on extensive context windows can lead to attention dilution and reduced focus on the original query [13,9]. Furthermore, the sequential processing approach struggles with multi-turn conversations [11] where context drift becomes problematic, and the stop-token mechanisms for API call extraction lack the robustness needed for integration with newer LLMs.

In response to these limitations and building upon recent advances in multi-agent LLM systems [4], we propose a novel multi-agent architecture that addresses these efficiency bottlenecks through specialized agent coordination and dynamic task decomposition. We first conduct a GeneGPT reproducibility study and adapt the system to more recent LLMs to identify key limitations. Second, we introduce `GenomAgent`, a multi-agent framework that extends GeneGPT's capabilities. Experimental results show `GenomAgent` achieves an average performance score of 0.93 ($+12\%$ over GeneGPT's 0.83) while reducing computational costs by 79% ($2.11 vs. $10.06 total) across the GeneTuring benchmark [8].

The remainder of the paper is organized as follows: Section 1 reviews GeneGPT, Section 3 details GeneGPT replication, Section 3 describes `GenomAgent`, Section 5 presents the experiments, and Section 6 provides some final remarks.

2   GeneGPT

GeneGPT[10] is a domain-specific system that enhances LLMs by integrating a tool-augmented architecture to connect Natural Language (NL) queries with structured genomics databases. It utilizes in-context learning , enabling the LLM to dynamically generate and execute API calls to external resources, thus allowing real-time data retrieval and synthesis. This approach overcomes the limitations of static knowledge repositories in pre-trained models and demonstrates the extended utility of LLMs in specialized fields by ensuring access to up-to-date, structured data, while retaining their NLP capabilities for scientific QA.

GeneGPT employs a specialized prompting strategy that leverages the code completion capabilities of LLMs. It is based on OpenAI Codex[6], and the prompt structure includes task instructions, relevant API documentation for E-utils and BLAST [17,2,7], in-context learning examples, and the target question. GeneGPT uses the special symbol "$\rightarrow$" as a stop token to identify API calls. When the LLM generates text containing this symbol, the system: (1) extracts the URL using a regex pattern; (2) executes the API call; and (3) appends the API result to the prompt. The model then continues generation, repeating steps 1-3 for any additional API calls, until the termination token "\n\n" is detected. Then, the LLM generates the final answer using the retrieved results and in-context understanding of the examples.

GeneGPT was developed in four configurations: full, slim, turbo, and lang. In full settings, the system incorporates complete API documentation and four examples, while slim uses only two examples. The turbo configuration replaces Codex with `GPT-3.5-turbo-16k`, and lang implements the ReAct framework [22]. The system was evaluated on nine tasks in the GeneTuring benchmark. Based on the experimental results, GeneGPT achieves state-of-the-art performance with an average performance score of 0.83, which substantially outperforms baselines as Bing Chat (0.44), BioMedLM [14] (0.08), and GPT-3 (0.16).

GeneGPT performance was assessed across multiple evaluation metrics designed for different tasks within the GeneTuring benchmark. These include exact match accuracy for nomenclature tasks, recall for association tasks, and task-

Table 1. Results of the reproducibility of GeneGPT on the GeneTuring Benchmark.

Model	Nomenclature		GenomicLocation			FunctionalAnalysis		SequenceAlignment	
	Gene Alias	Name Conv.	SNP Assoc.	Gene Loc.	SNP Loc.	Disease Assoc.	Protein Genes	DNA to Human	DNA to Species
GeneGPT Turbo	0.64	1.00	0.96	0.54	0.98	0.63	0.96	0.42	0.88
Reproduced	0.68	0.98	0.90	0.54	0.92	0.56	0.80	0.07	0.62
Relative diff	6.25%	-2.00%	-6.25%	0.00%	-6.12%	-11.11%	-16.67%	-83.33%	-29.55%
GeneGPT Lang	0.76	0.02	0.90	0.54	0.74	0.39	0.90	0.06	0.54
Reproduced	0.76	0.92	1.00	0.72	1.00	0.76	1.00	0.31	0.54
Relative diff	0.00%	4500%	11.11%	33.33%	35.14%	94.87%	11.11%	416.67%	0.00%

specific scoring for alignment tasks. While individual task metrics employ different evaluation criteria and cannot be directly compared inter-task due to varying task complexity and requirements, all metrics are normalized in $[0, 1]$, enabling uniform interpretation. For comparative analysis, following an established approach in multi-task evaluation [20,21], we report a macro-averaged performance score computed as the arithmetic mean across all task-specific metrics, providing a singular measure of overall system accuracy while acknowledging that this aggregate metric represents a simplified view of the system's diverse capabilities across heterogeneous genomics QA tasks.

3   Reproducibility of GeneGPT

To understand GeneGPT's operational principles and identify improvement opportunities, we conducted a reproducibility study. The original system relied on `code-davinci-002` and `GPT-3.5-turbo-16k`, which were deprecated in 2023 and 2024, respectively[3]. We selected `GPT-4o-mini` as the replacement model due to its performance, cost efficiency, and current stability. We implemented two compatible configurations: `turbo` and `lang`. The original paper for the `lang` setting mentions only *LangChain* as the orchestration framework without detailing its implementation. Due to substantial changes and deprecation in favor of LangGraph[4], we opted for LangGraph for this configuration. We preserve GeneGPT's core design based on the stop-token interaction mechanism.

During the reproduction process, we encountered two main challenges. First, GPT-4o-mini did not consistently follow the URL generation format required by GeneGPT's extraction pipeline. We addressed this by explicitly prompting the model to use the desired format. Second, the original implementation used context truncation to avoid exceeding length limits, which hindered HTML data extraction by discarding critical information. We removed this limit with GPT-4o-mini's larger context window. Unlike the original system's single-token outputs, the reproduced system often requires manual extraction from multi-sentence responses before automatic evaluation.
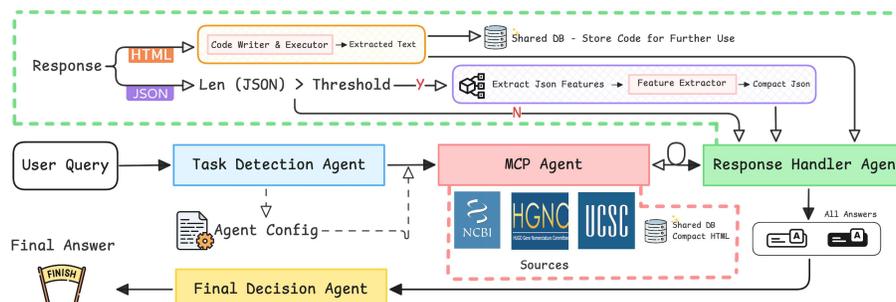
---

[3] https://platform.openai.com/docs/deprecations
[4] https://langchain-ai.github.io

Fig. 1. `GenomAgent` multi-agent architecture and workflow.

For the reproducibility analysis, we employ the GeneTuring Benchmark, which encompasses 12 distinct tasks, each comprising 50 question-answer pairs. We approached 9 of these GeneTuring tasks, replicating the original GeneGPT paper. These selected tasks are grouped into four main subcategories: (1) nomenclature inquiries, focusing on gene aliases and name transformations; (2) genomic location inquiries, examining the positioning of genes and SNPs and their interrelations; (3) functional analysis inquiries, investigating aspects such as gene-disease associations and the genes responsible for protein coding; and (4) sequence alignment inquiries, which involve mapping DNA sequences to the human genome and comparing them across various species.

Table 1 presents the reproduced results. Our reproduced system consistently shows improvements in the `lang` setting; these gains show that correct implementation of ReAct architecture with newer models can increase performance. However, in `turbo` settings, we observed high variation and degradation as a result of the non-compatibility of stop-token processing with general-purpose LLMs. We manually reviewed and categorized all the mistakes made by the system we replicated into three distinct types: E1: incomplete data coverage, where correct answers do not exist in NCBI; E2: stop-token parsing failures, where LLM does not generate API calls in the expected format; E3: context loss, where large API responses cause LLM to lose focus on the original question. Our results suggest that the reproduced `turbo` setting causes errors due to E2, where the system gets stuck in a loop, and ultimately, no results are achieved. In contrast, in `lang` mode, the most dominant errors are related to E1 and E3.

4   GenomAgent

We present `GenomAgent`, a multi-agent architecture (see Figure 1) that extends beyond single-agent approaches for biomedical QA. The system uses multiple specialized agents to handle questions through a coordinated workflow and enable flexible interaction with various biomedical APIs and DBs.

`GenomAgent` implements a hierarchical multi-agent architecture comprising four core processing agents and three specialized utility agents. The **Task De-**

tection Agent serves as the initial query router, performing intent classification to determine appropriate processing workflows based on predefined configuration schemas. The **Multi-source Coordination Protocol (MCP) Agent** orchestrates parallel API interactions across heterogeneous biomedical databases (NCBI [7], HGNC [18], UCSC [16]), implementing asynchronous query dispatch and response aggregation protocols. The **Response Handler Agent** processes heterogeneous API responses through dual processing pipelines: (1) JSON responses undergo threshold-based evaluation, triggering the **Feature Extractor Agent** for schema summarization when size limits are exceeded, and (2) HTML responses activate the **Code Writer Agent** to generate targeted extraction scripts executed by the **Code Executor Agent**. Generated extraction code is cached in a shared repository to enable reuse and reduce computational overhead. The **Final Decision Agent** performs multi-source response synthesis using consensus-based aggregation algorithms to generate coherent answers.

Built on the Google Agent Development Kit, `GenomAgent` addresses three critical limitations identified in GeneGPT: (1) *source diversity* through multi-database querying to reduce information gaps, (2) *modular processing* via specialized agents to handle heterogeneous response formats, and (3) *adaptive extraction* through dynamic code generation for complex data structures. This architecture enables parallel processing, reduces context window constraints, and provides fault tolerance through distributed task execution.

5 Experiments

`GenomAgent` evaluation follows the same experimental setup as our reproducibility study with enhanced precision improvements. Task specific evaluation metrics include: exact matching for nomenclature and genomic location tasks; recall calculation based on exact gene matches for gene-disease associations; vocabulary-mapped exact matching for cross-species DNA alignment (mapping Latin to common names, e.g., "Homo sapiens" to "human"); and partial scoring for human genome alignment, awarding 0.5 points for correct chromosome identification with incorrect positions (e.g., `chr8:708-882` vs. `chr8:120-121`).

Our experimental protocol differs from GeneGPT in two key aspects: (1) expanded vocabulary mappings to accommodate updated NCBI species annotations, and (2) enhanced partial scoring that calculates sequence-level similarity for both start and end positions in alignment tasks. We applied identical evaluation protocols to both GeneGPT and `GenomAgent` to ensure fair comparison. The expanded vocabulary mappings and partial scoring mechanisms were applied to both systems when evaluating on the GeneTuring benchmark.

Furthermore, we quantify the computational cost for each task. This is achieved by tracking the number of input and output tokens and applying real-model pricing to derive the total cost per task.

Table 2 reports the performance and cost of `GenomAgent` in GeneTuring tasks compared to GeneGPT's main results. `GenomAgent` achieves substantial improvements in both performance and computational efficiency. Our model attains an

Table 2. Performance and cost (\$) on GeneTuring. Best existing models are underlined; bottom row shows `GenomAgent`'s improvement over best baseline.

Model	Nomenclature		Genomic Location		Functional Analysis		Sequence Alignment		Overall	
	Score↑	Cost↓	Score↑	Cost↓	Score↑	Cost↓	Score↑	Cost↓	Avg↑	Total(\$)↓
GeneGPT Full	0.90	2.19	0.87	<u>2.36</u>	0.76	2.13	0.65	1.69	0.80	11.14
GeneGPT Slim	<u>0.92</u>	<u>1.63</u>	<u>0.88</u>	2.52	<u>0.84</u>	<u>1.67</u>	<u>0.66</u>	<u>1.74</u>	<u>0.83</u>	<u>10.06</u>
GeneGPT Turbo	0.82	4.39	0.82	4.73	0.80	4.25	0.65	3.39	0.78	16.76
GeneGPT Lang	0.39	n/a	0.73	n/a	0.65	n/a	0.30	n/a	0.54	n/a
GenomAgent (Ours)	**0.98**	**0.43**	**0.98**	**0.88**	**0.89**	**0.25**	**0.85**	**0.55**	**0.93**	**2.11**
Improvement (%)	6.5%	73.6%	11.4%	62.7%	4.8%	85.0%	28.8%	68.4%	12.0%	79.0%

Fig. 2. Performance-cost tradeoff on GeneTuring. Bubble size shows normalized cost; High Value Region shows optimal performance at minimal cost.

average score of 0.93, exceeding the best-performing GeneGPT model (0.83). In simple tasks (nomenclature and genomic location), our system achieves near-perfect performance with a score of 0.98, surpassing GeneGPT-slim's scores of 0.92 for nomenclature and 0.88 for genomic location. Most notably, in alignment tasks, which are the most challenging task for GeneGPT, we achieve a remarkable 28.8% improvement. Computational cost analysis reveals even more striking improvements. `GenomAgent` costs only \$2.11 total in all tasks (79.0% reduction from best-performing GeneGPT (\$10.06)).

In addition, as shown in Figure 2, `GenomAgent` is the optimal selection, as it achieves a high score at minimal computational expense.

6 Final Remarks and Future Work

In this study, we reproduce GeneGPT [10] to pinpoint three critical bottlenecks: (i) limited data coverage, (ii) parsing failures, and (iii) context loss in multi-turn queries. We then introduce `GenomAgent`, a hierarchical multi-agent framework that orchestrates parallel API queries, dynamic data extraction, and consensus-based response synthesis. Evaluated on the GeneTuring benchmark, `GenomAgent` achieves a 12% increase in average performance (0.93 vs. 0.83) and a 79% reduction in computational cost ($2.11 vs. $10.06). Sequence alignment tasks see the largest gains (28.8%), driven by multi-source retrieval and adaptive partial scoring. Unlike GeneGPT's rigid single-agent design, `GenomAgent`'s modular agents seamlessly adapt to new LLMs and evolving database schemas. These results demonstrate that coordinated multi-agent orchestration can deliver both superior accuracy and substantial resource efficiency for genomic question answering.

Looking ahead, our results suggest several promising research directions: First, we acknowledge that the 12% average improvement cannot be cleanly attributed to specific architectural choices without systematic ablation analysis. Decomposing components through controlled experiments that isolate individual elements can demonstrate the contribution of each architectural component. Second, our evaluation is limited to the GeneTuring benchmark. This restricted scope prevents us from fully validating `GenomAgent`'s generalizability across diverse genomic QA tasks. Third, investigating hybrid approaches that combine the efficiency of single-agent systems for simple queries with multi-agent coordination for complex tasks could optimize the performance-cost tradeoff. Fourth, the development of automated prompt optimization techniques for agent-specific instructions could further reduce the manual effort required for system configuration. Finally, extending our comparative analysis to include emerging state-of-the-art frameworks such as [5] will enable benchmarking of our system's capabilities against the latest advances. We will investigate all these dimensions in the planned future work.

Acknowledgments

Disclosure of Interests

The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ali, S., Qadri, Y.A., Ahmad, K., Lin, Z., Leung, M.F., Kim, S.W., Vasilakos, A.V., Zhou, T.: Large language models in genomics—a perspective on personalized medicine. Bioengineering **12**(5), 440 (2025)
2. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. Journal of Molecular Biology **215**(3), 403–410 (1990)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in Neural Information Processing Systems **33**, 1877–1901 (2020)
4. Cemri, M., Pan, M.Z., Yang, S., Agrawal, L.A., Chopra, B., Tiwari, R., Keutzer, K., Parameswaran, A., Klein, D., Ramchandran, K., et al.: Why do multi-agent LLM systems fail? arXiv preprint arXiv:2503.13657 (2025)
5. Chen, H., Zuccon, G., Leelanupab, T.: Beyond genegpt: A multi-agent architecture with open-source llms for enhanced genomic question answering. In: Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region. pp. 143–152 (2025)
6. Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H.P.D.O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al.: Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374 (2021)
7. Coordinators, N.R.: Database resources of the national center for biotechnology information. Nucleic Acids Research **43**(D1), D6–D17 (2015)
8. Hou, W., Ji, Z.: Geneturing tests gpt models in genomics. BioRxiv: The Preprint Server for Biology (2023)
9. Hsieh, C.Y., Chuang, Y.S., Li, C.L., Wang, Z., Le, L., Kumar, A., Glass, J., Ratner, A., Lee, C.Y., Krishna, R., et al.: Found in the middle: Calibrating positional attention bias improves long context utilization. In: Findings of the Association for Computational Linguistics: ACL 2024. pp. 14982–14995 (2024)
10. Jin, Q., Yang, Y., Chen, Q., Lu, Z.: Genegpt: Augmenting large language models with domain tools for improved access to biomedical information. Bioinformatics **40**(2), btae075 (2024)
11. Laban, P., Hayashi, H., Zhou, Y., Neville, J.: Llms get lost in multi-turn conversation. arXiv preprint arXiv:2505.06120 (2025)
12. Li, Q., Li, L., Li, Y.: Developing ChatGPT for biology and medicine: a complete review of biomedical question answering. Biophysics Reports **10**(3), 152 (2024)
13. Liu, N.F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., Liang, P.: Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics **12**, 157–173 (2024)
14. Luo, R., Sun, L., Xia, Y., Qin, T., Zhang, S., Poon, H., Liu, T.Y.: BioGPT: generative pre-trained transformer for biomedical text generation and mining. Briefings in Bioinformatics **23**(6), bbac409 (2022)
15. Masterman, T., Besen, S., Sawtell, M., Chao, A.: The landscape of emerging AI agent architectures for reasoning, planning, and tool calling: A survey. arXiv preprint arXiv:2404.11584 (2024)
16. Perez, G., Barber, G.P., Benet-Pages, A., Casper, J., Clawson, H., Diekhans, M., Fischer, C., Gonzalez, J.N., Hinrichs, A.S., Lee, C.M., et al.: The UCSC genome browser database: 2025 update. Nucleic Acids Research **53**(D1), D1243–D1249 (2025)
17. Schuler, G.D., Epstein, J.A., Ohkawa, H., Kans, J.A.: [10] entrez: Molecular biology database and retrieval system. In: Methods in Enzymology, vol. 266, pp. 141–162. Elsevier (1996)

18. Seal, R.L., Braschi, B., Gray, K., Jones, T.E., Tweedie, S., Haim-Vilmovsky, L., Bruford, E.A.: Genenames. org: the HGNC resources in 2023. Nucleic Acids Research **51**(D1), D1003–D1009 (2023)
19. Shen, Z.: LLM with tools: A survey. arXiv preprint arXiv:2409.18807 (2024)
20. Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: Superglue: A stickier benchmark for general-purpose language understanding systems. Advances in Neural Information Processing Systems **32** (2019)
21. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: Glue: A multitask benchmark and analysis platform for natural language understanding. In: Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP. pp. 353–355 (2018)
22. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K.R., Cao, Y.: React: Synergizing reasoning and acting in language models. In: The Eleventh International Conference on Learning Representations (2023)