Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Silent methodological failures in AI-assisted science: benchmarking large language model agents for virtual gene knockout analysis

Bo Wang
Second Affiliated Hospital of Xi'an Jiaotong University
Jun Li
Second Affiliated Hospital of Xi'an Jiaotong University
Shu Zhang
drzhangshu@xjtu.edu.cn

Second Affiliated Hospital of Xi'an Jiaotong University

Article

Keywords: Large language models, Computational biology, Silent methodological failures, Single-cell RNA-sequencing, Virtual gene knockout

Posted Date: November 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11399586/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

Large language models (LLMs) are increasingly deployed as autonomous agents in computational biology, yet their reliability in executing complex scientific workflows remains poorly characterized. Here, we benchmarked five LLM agents (DeepSeek V4 Pro, GLM 5.1, HY3 Preview, Kimi K2.6, and Minimax M2.7) on virtual gene knockout analyses using four computational frameworks across two single-cell RNA-sequencing datasets. All agents successfully completed the assigned workflows, but their quantitative outputs varied substantially: execution times spanned a 31fold range (61 – 1,920 s), while cross-model coefficients of variation reached 142.1%. Notably, four of five agents (80%) exhibited "silent methodological failures," generating computationally valid, error-free code that produced biologically implausible results. Among the evaluated agents, GLM 5.1 showed the most balanced performance across computational efficiency and biological plausibility. These findings demonstrate that successful code execution alone does not ensure scientific validity and highlight cross-model concordance and biological validation as important quality-control strategies for Artificial Intelligence (AI)-assisted computational biology workflows.

1 Introduction

The deployment of large language models (LLMs) as autonomous agents is reshaping scientific computing by enabling researchers to delegate the construction and execution of complex analytical pipelines to natural-language-driven systems(1–3). While this form of Artificial Intelligence (AI)-assisted science may broaden access to sophisticated computational workflows(4), it also raises important concerns regarding methodological fidelity and reproducibility. In particular, when independent AI agents receive the same analytical protocol, it remains unclear whether they will implement the intended algorithms consistently and produce computationally and biologically plausible results. This uncertainty is especially relevant in scientific applications, where executable code alone does not guarantee methodological fidelity or valid biological interpretation. Such concerns contribute to broader discussions about the reliability of AI-generated scientific outputs(5).

Systematic evaluations of LLM agents on end-to-end computational biology tasks remain limited. Existing benchmarks primarily assess code-generation performance on isolated programming problems, such as HumanEval, SWE-bench, and SciCode(1–3), rather than complete scientific workflows involving data parsing, algorithm implementation, parameter specification, and statistical interpretation. Emerging benchmarks for scientific AI agents and broader evaluations of LLM applications in bioinformatics provide important foundations(4, 6), but the reproducibility and methodological reliability of LLM-generated computational biology workflows across independently implemented agents remain insufficiently characterized(7).

Virtual gene knockout (in silico perturbation) provides a suitable testbed for evaluating this reliability. By computationally simulating gene perturbation in single-cell transcriptomic data and assessing downstream changes in gene regulatory networks (GRNs), these workflows require the integration of

linear algebra, graph-based analysis, representation learning, and statistical inference. We therefore evaluated four computational paradigms: scTenifoldKnk, based on tensor decomposition and manifold alignment(8); scPII, based on perturbation response scanning(9); GenKI, based on variational graph autoencoders(10); and Geneformer, a foundation model pretrained on approximately 30 million single-cell transcriptomes(11, 12). These frameworks represent distinct computational strategies and provide complementary settings for assessing whether LLM agents can faithfully translate scientific protocols into executable analyses(13).

ALB (albumin) was selected as the perturbation target because it is highly expressed in hepatocytes and is closely associated with well-characterized liver-specific transcriptional programs. Its high expression and established biological context provide a strong-signal setting for evaluating perturbation-related computational outputs. Accordingly, ALB was used as a biologically interpretable benchmark target rather than as an absolute ground-truth reference for all computational outcomes(14–16). This design enables assessment of whether model-generated perturbation results are consistent with established ALB-associated biology, while recognizing that generalizability to other genes and perturbation contexts remains to be established.

Here, we present a standardized benchmark of LLM agents executing first-pass, end-to-end computational biology workflows. Using the WorkBuddy framework as a common agentic execution environment, five LLM agents independently implemented four virtual knockout frameworks across two single-cell RNA-sequencing datasets targeting ALB. Performance was evaluated across four dimensions: speed, correctness defined as biological plausibility, robustness, and cross-model concordance. All evaluated agents completed the assigned workflows, yet substantial differences emerged in execution time, quantitative outputs, and biological plausibility. Four of five agents (80%) exhibited “silent methodological failures,” in which computationally executable, error-free implementations produced results that failed the predefined criteria for biological plausibility. These findings demonstrate that successful code execution alone does not establish scientific validity and highlight the value of cross-model concordance and independent biological assessment as complementary quality-control strategies for AI-assisted computational biology(17).

2 Methods

2.1 Benchmark design

We designed a benchmark comprising four virtual gene knockout methods, two single-cell RNA-sequencing datasets, and one perturbation target (ALB), resulting in eight tasks per LLM agent. The five LLM agents were selected from the models natively supported by the WorkBuddy framework, enabling standardized deployment and execution within a common agentic environment. WorkBuddy served as centralized middleware to programmatically dispatch an identical natural-language prompt to each agent, execute the autonomously generated code within an isolated runtime environment, and capture first-pass computational outcomes without iterative refinement or human intervention. All analyses were

performed on identical hardware consisting of an Intel Core i5-14490F CPU (@ 4.9 GHz), 16 GB DDR5 RAM, 8 GB GDDR7 GPU, and Windows 11 Pro.

2.2 Datasets

Two publicly available scRNA-seq datasets from the Gene Expression Omnibus (GEO) were used. GSE149614 is a hepatocellular carcinoma (HCC) dataset containing raw counts for 71,915 cells across 25,712 genes(18, 19); a preprocessed AnnData file comprising 3,000 cells and 5,000 highly variable genes was provided for memory-constrained analyses using SCANPY(20). GSE290298 contains processed counts for approximately 3,000 cells across 5,000 genes(21). Both datasets were provided as local files in compressed CSV/TXT or H5AD format to minimize network-related variability during execution. ALB expression was confirmed in both datasets: in GSE149614, ALB was detected in 72.8% of cells (mean log-normalized expression, 2.50), whereas in GSE290298 it was detected in 63.2% of cells (mean, 1.50).

2.3 Virtual knockout methods and statistical criteria

Four computational frameworks were evaluated using predefined implementation and interpretation criteria. These criteria were specified to enable consistent assessment of outputs generated by the five LLM agents while preserving method-specific analytical characteristics.

scTenifoldKnk(8): PCA-based GRN construction using the top three principal components and 500 highly variable genes, followed by ALB edge removal, manifold alignment using L2 distance, and Z-score calculation with false discovery rate (FDR) correction(22). A result was classified as significant when the ALB Z-score was > 5 with FDR-adjusted $p < 0.05$.

scPII(9): co-expression GRN construction using Pearson correlation with top-k = 100 thresholding, followed by perturbation response scanning after removal of ALB-correlated edges and permutation-based p-value estimation (100-1,000 bootstrap iterations). The five agents implemented distinct operationalizations of the impact score (GLM: efficiency × sensitivity; DeepSeek: eigenvector centrality; HY3: L2 norm; Kimi: normalized connectivity; Minimax: absolute edge-weight sum). These operationalizations are mathematically distinct and produce non-equivalent quantities; because no unified impact-score definition was imposed in the benchmark, direct quantitative comparison of these scores was limited. Only DeepSeek reported an explicit p-value, whereas the other implementations were interpreted according to their respective output criteria. The absence of a prespecified, standardized impact-score definition was treated as a methodological limitation of this benchmark.

GenKI(10): graph perturbation analysis based on construction of a co-expression adjacency matrix, removal of ALB-associated node connections, and measurement of the resulting structural change using the Frobenius norm (Δ%). A Δ% >5% was classified as significant, 2–5% as borderline, and < 2% as not significant. These thresholds were supported by prior benchmarks of GRN perturbation methods(16). No direct p-value is generated by this graph-theoretical formulation; interpretation was therefore based on the magnitude of structural change.

Geneformer-based virtual knockout(11, 12): Geneformer-generated gene embeddings were reduced to 50 dimensions using PCA. ALB was computationally removed to simulate virtual knockout, and original and perturbed representations were projected using the original PCA transformation. Perturbation effects were quantified by per-cell cosine similarity, followed by a one-sample t-test against the identity null (cosine = 1.0). Significance required both $p < 0.05$ and mean $\Delta\cos = (1 - \cos) > 0.05$. Borderline cases ($\Delta\cos = 0.04–0.07$) were assessed based on biological relevance. PCA-based dimensionality reduction followed established single-cell workflows(20, 23).

2.4 Evaluation metrics

Models were evaluated across four orthogonal dimensions. Speed: total and per-task wall-clock execution time. Correctness: biological plausibility, assessed using the predefined statistical criteria described above (significant / borderline / not significant) and evaluated against established ALB-associated biology. Robustness: technical errors and successful self-recovery events. Concordance: cross-model variability quantified as $CV = \sigma/\mu \times 100\%$.

2.5 Statistical analysis

All analyses were performed in Python 3.13 using NumPy (v1.26+), SciPy (v1.14+), pandas (v2.2+), and AnnData (v0.10+). Multiple testing was controlled using the Benjamini-Hochberg procedure where applicable.

3 Results

3.1 Computational efficiency and algorithmic bottlenecks

End-to-end execution times varied substantially among the evaluated agents, spanning a 31.5-fold range from 60.9 s for Minimax M2.7 to 1,920.0 s for Kimi K2.6 (Fig. 1a). Kimi's prolonged execution time was associated with an autonomous recovery process following an "out-of-memory" (OOM) exception that occurred when the agent attempted to ingest the complete 25,712 × 71,915 raw-count matrix. The other agents avoided this memory bottleneck by defaulting to preprocessed data objects. Excluding the 1,800-s recovery episode, GLM 5.1 was the fastest agent (241.9 s), followed by HY3 Preview (413.0 s) and DeepSeek V4 Pro (593.8 s).

Across methods, scPII represented the major computational bottleneck, accounting for 50–73% of total execution time across the evaluated agents (Fig. 1b). This pattern is consistent with the computational burden of Pearson correlation-based network reconstruction, which scales quadratically with the number of genes(24). In contrast, scTenifoldKnk consistently required less than 1 s per dataset, reflecting the computational efficiency of its dimensionality-reduction and network-analysis strategy(8).

3.2 Divergence in virtual knockout significance classification

Classification of ALB perturbation significance revealed substantial inter-model divergence across the evaluated workflows (Fig. 1c). For scTenifoldKnk, four of the five agents produced Z-scores substantially above the predefined significance threshold ($Z > 5$), with values around $Z \approx 20$. In contrast, Minimax produced markedly lower Z-scores (1.12 and − 0.26) and therefore failed to meet the predefined significance criterion in these tasks.

Distinct patterns also emerged across the four computational frameworks. GLM 5.1 identified strong perturbation signals in network- and pathway-based analyses but showed minimal Geneformer embedding shifts, with a cosine-distance change of less than 0.01. DeepSeek V4 Pro produced substantial Geneformer embedding changes but did not meet the predefined scPII significance criterion in the two evaluated tasks ($p = 0.13$ and 1.0). This discrepancy was associated with its use of eigenvector centrality as the scPII impact-score operationalization rather than the edge-weight perturbation formulation used by other implementations. Kimi K2.6 classified the Geneformer GSE290298 result as borderline ($\Delta\cos = 0.05$), illustrating how method-specific thresholds can influence the classification of relatively small perturbation effects.

3.3 Quantitative divergence across algorithmic paradigms

For scTenifoldKnk, ALB perturbation Z-scores showed substantial agreement among four of the five evaluated agents, with values ranging from 17.5 to 22.3 and consistently exceeding the predefined significance threshold of $Z > 5$ (Fig. 2a). Minimax was a marked outlier, producing near-baseline Z-scores of 1.12 for GSE149614 and − 0.26 for GSE290298. This deviation was associated with an apparent transposition of the input data matrix, which altered the downstream PCA-derived representation. Although Kimi produced Z-scores within the range observed for the other agents, it classified all 500 tested genes as statistically significant, suggesting a potential issue in its implementation of Benjamini–Hochberg FDR correction(22).

The scPII framework exhibited the greatest inter-model quantitative variability. This divergence reflected the distinct operationalizations of the impact score used by the five agents: GLM used efficiency × sensitivity based on the canonical scPRS formulation; DeepSeek used eigenvector centrality; HY3 calculated an L2-norm difference; Kimi used a normalized connectivity measure; and Minimax calculated the absolute sum of edge weights. Because these definitions quantify different mathematical quantities, their numerical outputs are not directly interchangeable. This methodological heterogeneity provides a plausible explanation for the cross-model variability observed for scPII and is consistent with the sensitivity of GRN-based perturbation analyses to network construction and scoring methodology(9, 24).

GenKI showed moderate cross-model consistency, accompanied by dataset-dependent differences. For GSE149614, Frobenius norm changes ranged from 5.8% to 6.8% for DeepSeek, GLM, and Kimi, whereas HY3 produced a value near zero ($\Delta \approx 0.1\%$). In GSE290298, the HY3 result increased to 5.0% (Fig. 2a). Minimax consistently produced substantially smaller changes across both datasets ($\Delta \approx 0.3\%$). This pattern was associated with its construction of correlation matrices with zero diagonals, which affected the matrix structure following ALB removal.

The Geneformer-based virtual knockout showed a distinct pattern characterized by relatively high cross-model agreement within individual datasets but marked sensitivity between datasets (Fig. 2a). In GSE149614, DeepSeek, HY3, Kimi, and Minimax produced closely clustered perturbation changes (Δcos ≈ 0.33–0.35), indicating substantial changes in the Geneformer-derived cell representations following computational ALB removal. In GSE290298, these four agents showed a coordinated reduction in Δcos toward the predefined threshold. This dataset-dependent shift indicates that the magnitude of the inferred perturbation was sensitive to the input dataset. GLM remained a marked outlier across both datasets, producing Δcos values close to zero. Its implementation aggregated gene-level PCA embeddings into cell-level representations using weighted averaging, which may have diluted the contribution of ALB to the resulting cell representation. Although the corresponding one-sample t-test could reach statistical significance, the small effect size failed to satisfy the prespecified Δcos threshold, illustrating the importance of evaluating effect magnitude alongside statistical significance(20, 22).

3.4 Cross-model concordance varies by method and dataset

Cross-model CV analysis revealed substantial method-specific differences in quantitative concordance across the two datasets (Fig. 2b). scTenifoldKnk showed CVs of 53.3% and 57.1%, whereas scPII showed the highest variability, with CVs of 138.3% and 142.1%. The high scPII variability was primarily associated with the divergent impact-score operationalizations described above. Geneformer showed intermediate CVs of 55.7% and 57.8%, despite the presence of a marked GLM outlier. With the exception of GenKI, GSE290298 generally produced slightly higher CV values than GSE149614. This difference coincided with lower ALB expression in GSE290298 (63% versus 73% of cells), which may contribute to a weaker perturbation signal(1, 13). Overall, the results indicate that frameworks with more explicitly specified mathematical procedures, such as scTenifoldKnk, showed lower implementation variability than the less standardized scPII impact-score formulation within this benchmark.

Despite substantial variation in quantitative perturbation metrics, the top five affected genes showed consistent biological patterns across the evaluated implementations (Table 1). ALB, GC, APOH, TTR, and ORM1 were repeatedly identified among the most affected genes and represent liver-enriched or hepatocyte-associated secreted proteins. This pattern is consistent with established hepatocyte-specific expression and co-regulation programs(18).

Table 1
Top-5 genes most affected by ALB virtual knockout, identified consistently across all five LLMs.

Gene	Full name	Function	Rank	Biological rationale
ALB	Albumin	Hepatocyte secreted protein	1 (self)	Knockout target; central hub of hepatocyte GRN
GC	Vitamin D binding protein	Plasma vitamin D transport	2	Co-expressed with ALB in hepatocytes; shared regulatory elements
APOH	Beta-2-glycoprotein 1	Phospholipid binding / coagulation	3	Liver-enriched secreted protein; co-regulated ALB network node
TTR	Transthyretin	Thyroid hormone / retinol transport	4	ALB-correlated hepatocyte-specific secretory protein
ORM1	Orosomucoid 1	Acute phase reactant	5	Liver-specific; co-expressed with ALB in HCC

3.5 Silent methodological failures predominate over technical errors

A marked asymmetry was observed between explicit technical errors and silent methodological failures (Fig. 3a). Only Kimi encountered an explicit technical error, an out-of-memory condition that was subsequently resolved through an autonomous change in data source. In contrast, four of the five evaluated agents exhibited at least one silent methodological failure within the benchmark. These cases were characterized by computationally executable and error-free workflows that nevertheless produced results failing the predefined criteria for biological plausibility. Representative examples included input-data transposition in Minimax during scTenifoldKnk, altered matrix structure following Minimax's GenKI implementation, non-standard impact-score operationalization associated with false-negative scPII classifications in DeepSeek, and a small-effect-size statistical artifact in GLM's Geneformer implementation. Because these failures do not necessarily generate execution errors, they may remain undetected by conventional code-level quality checks and therefore represent an important reliability concern for autonomous scientific workflows(5, 25).

Biological plausibility was assessed using the predefined method-specific quantitative criteria together with established ALB-associated biology. Across the eight benchmark tasks, GLM achieved the highest biological plausibility rate (6/8, 75%), followed by HY3 (5/8, 62.5%), DeepSeek and Kimi (4/8 each, 50%), and Minimax (1/8, 12.5%) (Fig. 3b). These results indicate substantial heterogeneity in biological plausibility across the evaluated agents, with no single agent achieving uniformly plausible results across all four computational frameworks and both datasets.

3.6 Multidimensional performance assessment

Radar chart analysis across the four evaluation dimensions—correctness, speed, concordance, and robustness—revealed distinct performance profiles among the evaluated agents (Fig. 3c). GLM exhibited the most balanced overall profile (mean score = 7.8), with scores of 8.5 for speed, 8.0 for correctness, 7.5 for concordance, and 7.0 for robustness. Minimax showed the lowest overall profile (mean score = 3.8), despite achieving the highest speed score (10.0), with scores of 1.0 for correctness and 2.0 for both robustness and concordance. Kimi achieved the highest robustness score (9.0), reflecting its successful autonomous recovery from the OOM error, but received a lower speed score (2.0) because of the associated recovery time. These results illustrate the trade-offs among computational efficiency, correctness, robustness, and concordance and support multidimensional evaluation of LLM agents beyond single-axis code-execution benchmarks.

3.7 Architectural paradigms and computational efficiency

The generated implementations also differed substantially in their software architecture and computational execution strategies. GLM 5.1 and Kimi K2.6 adopted modular, method- and dataset-specific implementations, comprising 9 and 11 scripts, respectively, whereas DeepSeek, HY3, and Minimax used more monolithic implementations consisting of 1–3 scripts and 458-1,352 lines of code. GLM and Kimi used vectorized Pearson correlation calculations that required less than 2 s for 5,000-gene matrices, whereas HY3 and DeepSeek used batched approaches that were approximately 100-fold slower for GRN construction. These differences illustrate the substantial effect of implementation strategy on computational performance in single-cell workflows, where vectorized operations can substantially reduce the computational cost of large matrix calculations.

Notably, Kimi's scPII implementation was both the smallest and fastest among the evaluated implementations, comprising 139–182 lines of code and requiring approximately 1.0 s for execution. Thus, within this benchmark, code length did not directly predict execution time, highlighting the greater importance of algorithmic and implementation-level choices for computational efficiency.

4 Discussion

Our benchmark demonstrates substantial heterogeneity in the first-pass execution of computational biology workflows by the evaluated LLM agents. Although all five agents completed the assigned tasks without persistent execution failure, their execution times and quantitative perturbation outputs varied considerably, with execution time spanning 31.5-fold and cross-model CV reaching 142.1%. These differences indicate that successful code execution does not necessarily imply consistent implementation of an analytical protocol or agreement in quantitative results. In this benchmark, the principal source of variability was therefore not persistent code failure alone, but differences in how agents translated the same high-level protocol into specific computational implementations.

A major manifestation of this gap was the occurrence of silent methodological failures. Four of the five evaluated agents produced at least one computationally executable output that failed the predefined criteria for biological plausibility. These failures included data-handling errors, such as input-matrix

transposition; model-specification differences, such as altered matrix structure or cell-level representation aggregation; and statistical or metric-related issues, such as non-standard impact-score operationalizations and discrepancies in multiple-testing implementation. Such errors are difficult to detect through conventional code-execution or syntax-based checks because the resulting workflows can complete successfully while producing scientifically misleading outputs. This distinction between executable correctness and biological plausibility is particularly important for autonomous scientific workflows, in which downstream users may not inspect every implementation detail before interpreting the resulting analyses(5, 17).

The scPII results further illustrate the consequences of underspecified methodological instructions. The five agents implemented substantially different operationalizations of the impact score, including efficiency × sensitivity, eigenvector centrality, L2 norm, normalized connectivity, and absolute edge-weight sum. Because these quantities are mathematically non-equivalent, their numerical outputs cannot be interpreted as directly comparable measurements of the same effect. We refer to this pattern as underspecification amplification, whereby relatively minor ambiguity in a high-level protocol can lead to substantially different downstream implementations and quantitative outputs. This observation is consistent with previous work showing that GRN inference and perturbation analyses can be sensitive to network construction and parameterization(9, 24). For agent-assisted computational biology, the finding emphasizes the importance of specifying not only the analytical objective but also the mathematical operations, input representations, thresholds, and output definitions required to reproduce the intended method. More explicit protocol specifications may therefore reduce implementation variability and improve reproducibility, consistent with the broader emphasis on protocol rigor in scientific AI benchmarks(4).

The multidimensional evaluation also revealed distinct performance profiles among the evaluated agents, rather than a uniform pattern of strengths and weaknesses. Within this benchmark, GLM 5.1 showed the most balanced overall performance and achieved the highest biological plausibility rate (6/8 tasks), while HY3, DeepSeek, and Kimi showed intermediate rates and Minimax showed the lowest rate (1/8 tasks). However, these results should be interpreted as benchmark-specific performance profiles rather than as a general ranking of LLM capability. For example, DeepSeek produced substantial Geneformer-derived embedding changes but failed to meet the predefined scPII significance criterion in two tasks, illustrating that strong performance in one analytical framework does not necessarily translate across computational paradigms. Kimi successfully recovered from an OOM error but incurred a substantial time penalty, whereas Minimax achieved the shortest overall execution time but showed poor biological plausibility and concordance. These findings emphasize that computational efficiency, biological plausibility, robustness, and cross-model concordance represent partially distinct dimensions of agent performance.

The observation of silent methodological failures has practical implications for the deployment of LLM agents in computational biology. First, cross-model verification can serve as a useful quality-control strategy for critical analyses. Agreement across independently generated implementations may increase

confidence in a result, whereas substantial disagreement can identify workflows requiring additional inspection. Importantly, however, cross-model concordance should be regarded as a diagnostic measure rather than an independent ground truth. Second, algorithmic transparency and protocol specification are important for reproducibility. Protocols intended for autonomous execution should define key mathematical operations, data representations, thresholds, and output metrics rather than relying exclusively on high-level descriptions. Third, domain-expert review remains important for biological interpretation, particularly when computational outputs satisfy technical execution criteria but produce small, ambiguous, or method-dependent perturbation effects. Together, these measures provide complementary safeguards against silent failures in AI-assisted scientific workflows(5, 25).

Several limitations should be considered when interpreting these findings. First, the benchmark included five LLM agents, four virtual knockout frameworks, two datasets, and a single perturbation target, ALB. The results therefore characterize reliability profiles under this specific benchmark configuration rather than establishing general performance characteristics of LLM agents. In particular, the high expression and well-characterized biological context of ALB provide a relatively strong-signal setting; whether similar failure patterns occur for lower-expressed genes or weaker perturbation signals remains unknown and requires future validation across a larger target-gene panel. Second, all analyses were performed in a standardized commodity computing environment. This configuration ensured comparable execution conditions across agents but may not represent performance under GPU-optimized or higher-memory environments. Third, the benchmark evaluated first-pass outputs without iterative refinement or human intervention. This design was intentional, allowing direct comparison of how agents interpret an identical protocol, but performance may differ under iterative prompting, debugging, or optimized prompt engineering. Fourth, LLM outputs are inherently stochastic, and replicate runs may produce different implementations or results. Repeated-run variability was not systematically evaluated here and represents an important direction for future studies.

5 Conclusion

In this benchmark, substantial differences in quantitative outputs and biological plausibility were observed across independently implemented workflows, with silent methodological failures occurring in four of five evaluated agents despite error-free execution. This demonstrates that LLM agents can execute sophisticated computational biology workflows, but executable output alone is insufficient evidence of methodological or biological reliability. These findings highlight the need to evaluate AI-generated scientific workflows beyond code execution, incorporating explicit algorithmic specifications, cross-model concordance, and independent assessment of biological plausibility.

Abbreviations

LLM, large language models; AI, Artificial Intelligence; GRNs, gene regulatory networks; ALB, albumin; GEO, Gene Expression Omnibus; HCC, hepatocellular carcinoma; FDR, false discovery rate; CV, coefficient of variation.

Declarations

Ethics approval

Not applicable

Consent for publication

Not applicable

Competing interests

The authors declare that they have no competing interests.

Funding

This study received support from the Innovation Project for Medical Integration in XJTU (YXJLRH2022062). The funders played no role in the study design, data collection, analysis, or manuscript preparation.

Author Contribution

Bo Wang: Conceptualization, Writing-original draft, Formal analysis, Investigation, Visualization, Writing-review & editing. Jun Li: Conceptualization, Supervision, Writing-review & editing. Shu Zhang: Conceptualization, Supervision, Writing-review & editing, Funding acquisition.

Data Availability

The datasets analysed during the current study are available in the National Center for Biotechnology Information (https://www.ncbi.nlm.nih.gov/).

References

1. Chen, M. et al. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:210703374* (2021).
2. Jimenez, C. E. et al. (eds) Swe-Bench: Can Language Models Resolve Real-World Github Issues? *International Conference on Learning Representations*; (2024).
3. Tian, M. et al. Scicode: A Research Coding Benchmark Curated by Scientists. *Adv. Neural. Inf. Process. Syst.* **37**, 30624–30650 (2024).

4. Chen, Z. et al. (eds) Scienceagentbench: Toward Rigorous Assessment of Language Agents for Data-Driven Scientific Discovery. *International Conference on Learning Representations*; (2025).
5. Messeri, L. & Crockett, M. J. Artificial Intelligence and Illusions of Understanding in Scientific Research. *Nature* **627** (8002), 49–58 (2024).
6. Mitchener, L. et al. Bixbench: A Comprehensive Benchmark for Llm-Based Agents in Computational Biology. *arXiv preprint arXiv:250300096* (2025).
7. Wang, Z. et al. (eds) Large Language Models in Bioinformatics: A Survey. *Findings of the Association for Computational Linguistics: ACL* ; 2025. ; 2025. (2025).
8. Osorio, D. et al. Sctenifoldknk: An Efficient Virtual Knockout Tool for Gene Function Predictions Via Single-Cell Gene Regulatory Network Perturbation. *Patterns* **3**(3). (2022).
9. Gupta, S., Romero, S. & Cai, J. J. Mapping Gene Impact on Single-Cell Transcriptomic Networks Via Perturbation Response Scanning. *bioRxiv* :2025.12. 15.694358. (2025).
10. Yang, Y. et al. Gene Knockout Inference with Variational Graph Autoencoder Learning Single-Cell Gene Regulatory Networks. *Nucleic Acids Res.* **51** (13), 6578–6592 (2023).
11. Theodoris, C. V. et al. Transfer Learning Enables Predictions in Network Biology. *Nature* **618** (7965), 616–624 (2023).
12. Zhang, Y., Venkatesh, M. S. & Theodoris, C. V. Discovery of Candidate Therapeutic Targets with Geneformer. *Nature Protocols* :1–24. (2026).
13. Wei, Z. et al. Benchmarking Algorithms for Generalizable Single-Cell Perturbation Response Prediction. *Nat. Methods*. **23** (2), 451–464 (2026).
14. Dixit, A. et al. Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell Rna Profiling of Pooled Genetic Screens. *cell* 167(7):1853-66. e17. (2016).
15. Lotfollahi, M., Wolf, F. A. & Theis, F. J. Scgen Predicts Single-Cell Perturbation Responses. *Nat. Methods*. **16** (8), 715–721 (2019).
16. Peidli, S. et al. Scperturb: Harmonized Single-Cell Perturbation Data. *Nat. Methods*. **21** (3), 531–540 (2024).
17. Huang, L. et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Trans. Inform. Syst.* **43** (2), 1–55 (2025).
18. Lu, Y. et al. A Single-Cell Atlas of the Multicellular Ecosystem of Primary and Metastatic Hepatocellular Carcinoma. *Nat. Commun.* **13** (1), 4594 (2022).
19. Jiang, S. et al. Characterization of the Distinct Immune Microenvironments between Hepatocellular Carcinoma and Intrahepatic Cholangiocarcinoma. *Cancer Lett.* **588**, 216799 (2024).
20. Wolf, F. A., Angerer, P. & Theis, F. J. Scanpy: Large-Scale Single-Cell Gene Expression Data Analysis. *Genome Biol.* **19** (1), 15 (2018).
21. Lu, X. et al. Multi-Omics Profiling Reveals Downregulated Tumor Lysine Metabolism Reshaping the Immune Microenvironment and Therapeutic Responses in Hepatocellular Carcinoma. *J. Translational Med.* **23** (1), 1117 (2025).

22. Benjamini, Y. & Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *J. Roy. Stat. Soc.: Ser. B (Methodol.)*. **57** (1), 289–300 (1995).
23. Virshup, I. et al. The Scverse Project Provides a Computational Ecosystem for Single-Cell Omics Data Analysis. *Nat. Biotechnol.* **41** (5), 604–606 (2023).
24. Pratapa, A. et al. Benchmarking Algorithms for Gene Regulatory Network Inference from Single-Cell Transcriptomic Data. *Nat. Methods*. **17** (2), 147–154 (2020).
25. Ziemann, M., Poulain, P. & Bora, A. The Five Pillars of Computational Reproducibility: Bioinformatics and Beyond. *Brief. Bioinform.* **24** (6), bbad375 (2023).