BEYOND BENCHMARK SCORES: AN ANALYSIS OF BIOINFORMATICS AGENT EVALUATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

As AI agents become capable of performing complex biological analyses, evaluating what they actually understand has become as important as evaluating what they can do. Bioinformatics benchmarks have followed this progression, moving from question answering towards evaluating if agents can conduct open-ended computational biology research tasks. This increased realism comes at the cost of verifiability. As more of the scientific process is delegated to the agent, it is not enough to evaluate just the results of the analysis, but to understand if we can trust the scientific process through which the agent arrives at the results. To get to the bottom of this question, this paper tries to systematically analyze components of a good evaluation and how to find out if we're actually evaluating biological reasoning.

1 INTRODUCTION

Agentic systems are becoming increasingly capable of executing biological data analysis workflows. Practitioners use agents to inspect sequencing files, choose preprocessing pipelines, apply bioinformatics algorithms to data, interpret assays, search literature, reason across modalities, design experiments, and even to interact with laboratory equipment (Fa et al., 2026; Qu et al., 2026; Alber et al., 2026; Bu et al., 2026; Ghareeb et al., 2026; Qu et al., 2025; Qiu et al., 2025). This expansion in capability necessitates benchmarks that reliably measure these increasingly complex forms of competence.

Early evaluations of Large Language Models (LLMs) for biology tested question answering (Wang et al., 2024), while more recent evaluations (Mitchener et al., 2025) require models to perform multiple parts of the scientific workflow to obtain an answer (e.g., writing data processing code, selecting the appropriate analysis, choosing appropriate reference data).

Subsequent benchmarks moved into a regime that operates on minimally processed biological data and requires agents to carry out long-horizon analyses that involve substantial compute, extended execution times and interaction with complex software environments (Fa et al., 2026; Panigrahi et al., 2026; Diks et al., 2026b; Qu et al., 2026; Li & Ho, 2026). The agent is expected to plan the analysis path, install and manage packages, obtain reference data, execute tools, revise analytical choices and finally, integrate evidence into a scientifically defensible conclusion.

With agents' capabilities in executing long-horizon workflows increasing, we should bring the evaluations closer to realistic use cases. However, improved realism of a testbed comes at a cost of reduced verifiability. As we try to both develop increasingly autonomous systems and increase the task scope, we give agents greater freedom, whereby their performance becomes harder to evaluate in practice because the space of variables to measure becomes more heterogeneous and complex. Clearly, an agent may achieve a perfect score on a benchmark that probes biological knowledge through multiple-choice question answering but still fail when required to execute multi-step workflows that require applying that knowledge in practice.

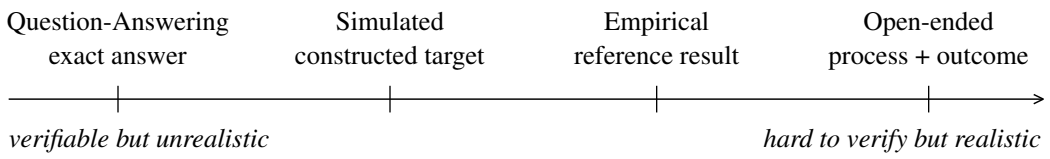
More constrained tasks are easier to verify, but less representative of both research and industrial workflows. Furthermore, such benchmarks ignore the reasoning process behind the final output entirely, thereby obscuring the causes of failure modes. Therefore, while providing trustworthy grounds for capability comparison between systems, these benchmarks supply limited information to practitioners about the avenues for improvement. On the other hand, evaluating the performance of

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

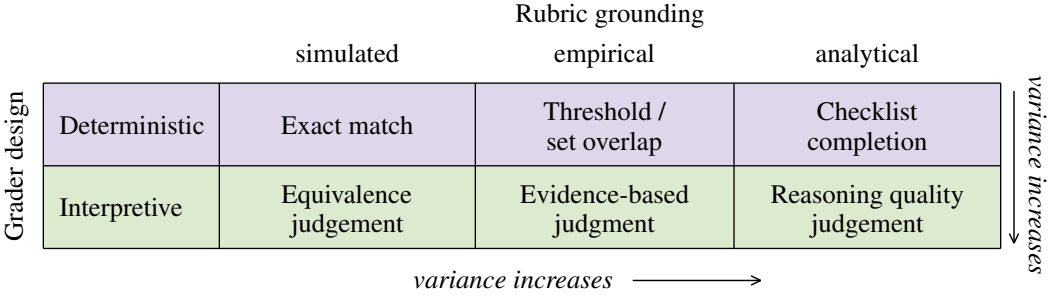
agents on realistic tasks requires relaxing success criteria to account for a larger plausible solution space. In these setups, metrics relying on exact match between the ground truth and an agent's outputs are no longer useful as enumerating all the plausible solutions is not tractable. Evaluating the agent performance on completing realistic pipelines thus requires coordinating expensive human or LLM-judges.

An ideal benchmark maximizes realism while retaining programmable verification and practical compute and time requirements for reliable statistical estimates (Figure **??**).

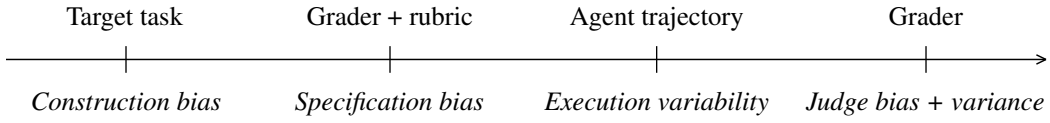
a) evaluation design is a trade-off between verifiability and realism



b) score reflects design choices which come with measurement error



Where error enters the score



c) agent failure modes and diagnostic tests

Failure mode	Diagnostic
General	Trajectory inspection
Data and grounding	Adversarial perturbations
Analysis / workflow	Multiple runs under varying configurations
Statistical inference	Simulated targets, controlled interventions
Interpretation	Counterfactuals

Figure 1: Paper overview. a) realism trades off against scalable programmable verification, b) target grounding and grader design jointly determine both the scoring rule and its characteristic errors. Variance in evaluations increases with rubric expressivity and by moving from deterministic to interpretive graders. c) Diagnosing different failure modes requires developing specialized probes.

2 PECULIARITIES OF AGENTIC EVALUATIONS IN BIOINFORMATICS

Generally, agentic evaluations are more difficult than single-turn ones because they're composed of a sequence of interactions with the data, tools and the environment. Along with evaluating if the final outcome is correct, agentic evals also need to determine which aspects of the trajectory, environment state, agent harness, the grader and its rubric are relevant to determining success (Anthropic, 2026). So, even in more verifiable domains, such as software development, there is still contention over the relationship between benchmark performance and real-world usability (Garg et al., 2026).

Besides the verifiability of biology as a domain itself, if we strip down the evaluations to just pure tool execution on biological data, evaluations in biology have much higher computational requirements than those of general-purpose agentic benchmarks. Answering *which genes are active in each cell, and which cell populations are present?* requires processing raw single-cell sequencing data for which a representative 20,000 cell analysis consumes 72.7 CPU-core-hours and 64GB of RAM, rising to 562 core-hours and 512GB of RAM for one million cells (10x Genomics, Cell Ranger System Requirements). Thus, a biological tool call may require hours to hundreds of CPU-hours of computation before higher-level inference on the data even begins.

Such datasets are orders of magnitude larger than the textual context available to AI agents and therefore cannot simply be supplied to the context of a single API call to an LLM grader, or extensively reviewed by a human grader in a short amount of time. Due to the sheer amount of data generated, even just tracing a model's trajectory is not a trivial problem.

Biological insights are often conditional on analytical and statistical inference design choices for which there is no single mechanically verifiable answer. A significant statistic on a single dataset is not enough for reliable verifiability (Fa & Culjak, 2026), which is often used as a rubric for grounding criteria. Evaluating whether an analysis is correct, in an ideal scenario, would involve having access to the experimental design itself, which is almost never the case in current evaluations.

These design choices are not purely philosophical considerations, as they inform grader and rubric design explicitly, which in turn guides post-training of agentic systems on these tasks and their real-world characteristics and performance. In the following section, we will therefore analyze rubric and grader designs. We build our analysis around the following question: *If there are not enough verifiable problems in biology (in the context of both training and evaluation signal) and we have to lean on familiar, expert-designed rubrics and grading, how do we measure whether there is any underlying reasoning rather than just doing familiar analysis on the data?*

3 GRADER AND RUBRIC DESIGN

A key part of developing a high-fidelity evaluation suite is determining both what should be assessed and how that assessment should be carried out. **Rubric** design defines the criteria used to evaluate the quality of a response, while **grader** design determines how those criteria are applied to produce a score. **Score** is the value produced by the **grader** when it applies the criteria defined in the **rubric** to an outcome and/or a trajectory.

The definition of a reported score in benchmarks varies widely. With respect to score granularity, the absolute score can be either **binary**, **graded**, or **continuous**. A binary signal assigns one of two outcomes, such as pass or fail. A graded signal uses several fixed levels, such as a score from 1 to 5, while a continuous signal takes a real value within a range, such as between 0 and 1. Reported scores can also be relative. A **preference** signal compares multiple responses and indicates which one is better, rather than assigning an absolute score to each response. Absolute scores are efficient to compute, but their reliability is unclear, as they can only be produced by an LLM judge. On the other hand, preference scores are more reliable but can be prohibitively expensive to compute.

Rubric and grader design are more interesting from our perspective, and these can take multiple forms and combinations. The common property of the rubrics is that so far humans have been the goal setters, and they decide what outcome the benchmark should reward. The rubric therefore specifies what evidence will count as success.

Another important component of an evaluation is the **grounding** of the rubric itself. In particular, the notion of a "correct" outcome is not easily defined in biology, as we have noted earlier. However,

we have to make some concessions and try to map the messy biology to practical agentic training and evaluation. To that end, from current work, we see an emergent systematization of rubrics. As summarized in Table 1, most **rubrics** have been either: (a) empirical reference, (b) deliberately constructed targets, or (c) inferred process targets.

Table 1: Bases for constructing evaluation targets.[a]

Basis	Description	Examples
Empirical	Derived from an existing empirical reference, such as a published or independently reproduced biological result.	SpatialBench (Workman et al., 2026b), scBench (Workman et al., 2026a), EpiBench (Muralidharan et al., 2026), VariantBench (Bhowmick et al., 2026), SpatialBench-Long (Diks et al., 2026a), and scBench-Long (Diks et al., 2026b); HeurekaBench (Panigrahi et al., 2026).
Simulated	Based on a simulated or otherwise constructed target result, including synthetic or augmented ground truths that can be verified directly.	CompBioBench (Nair et al., 2026), ScienceBoard (Sun et al., 2025), and GeneBench-Pro (Li & Ho, 2026).
Analytical	Rewards completion of the analytical steps considered appropriate for the task.	BiomniBench (Qu et al., 2026) and BioAgent Bench (Fa et al., 2026).

[a] These categories are not mutually exclusive. For the purposes of the taxonomy used in this paper, each benchmark is assigned to the category that most distinctly characterizes its evaluation target in relation to the ideas developed here.

Grader design is more straightforward to classify. In practice, we observe two broad directions: deterministic and interpretive graders. Their properties and representative examples are summarized in Table 2.

Table 2: Grader designs used to apply evaluation rubrics.

Grader design	Description	Examples
Deterministic	Uses programmatic checks that return the same score for the same output. This is reproducible but may reject valid answers outside predefined vocabularies or numerical tolerances.	SpatialBench (Workman et al., 2026b), CompBioBench (Nair et al., 2026), and GeneBench-Pro (Li & Ho, 2026).
Interpretive	Uses a human or LLM judge to assess rubric satisfaction. This accommodates scientifically defensible variation but introduces variability across judges and repeated evaluations.	BiomniBench (Qu et al., 2026) and SpatialBench-Long (Diks et al., 2026a).

Rubric design and grader choice form a matrix of evaluation strategies (see Figure **??**b). Moving from deterministic to interpretive grading generally increases flexibility but also introduces greater judge-dependent variance. In the same way, moving from simulated targets toward inferred analytical targets increases variance and makes the task less deterministic.

A common weakness of purely outcome-driven prompts is task underspecification, which can lead to artificially low evaluation scores. In some evaluations, only the desired outcome is specified, while the agent is also judged on its underlying reasoning or execution trajectory. [1]

There exists an optimal point between task specification [2] and interpretability of the grader and rubric design. In practice, if the task is overspecified and the underlying biology is highly verifiable,

[1] Imagine instructing an agent to *"build a working audio-processing application"*, while secretly grading whether it uses a particular FFT implementation, threading model, and file-decoding pipeline.

[2] Prompt that defines the goal of the analysis

deterministic grading can be the optimal choice. However, if the task is underspecified, it is beneficial to introduce flexibility to the grader to allow for fairer evaluation.

In practice, we see examples of graders using all of the classes of rubrics and combining them into one for the grading process. This serves to mitigate the drawbacks of each of these rubrics.

The base case from which a researcher might start is the most obvious case of an **empirical reference target** with a programmatically verifiable outcome using a **deterministic grader**. A deterministic grader has low conditional variance; given the same output, it produces the same score, but at the cost of specification bias. Its numerical tolerances and anticipated outcome set define a fixed acceptance surface. Consequently, it may reject a scientifically defensible answer that its designers did not anticipate.

An audit of SpatialBench (Workman, 2026) identified two recurring problems. Task ambiguity, where prompts left analytical decisions unspecified, and threshold sensitivity, where numerical tolerances were too narrow to accommodate valid analysis paths. In one task, both the reference solution and an independent expert recovered the intended conclusion that two genes were strongly co-expressed, but obtained correlations of (0.62) and (0.85), due to different normalization methods. The expert's valid answer fell outside the grader's acceptance range. Six domain experts independently attempted all 159 tasks using only the prompt and data, yet only 94 tasks (59.1%) passed the original graders in the first review round.

Even apparently simple, verifiable tasks are subject to such specification errors. BioAgent Bench (Fa et al., 2026) instructs an agent to identify viruses in a dolphin fecal sample and return a CSV containing contig counts and taxonomic labels. A deterministic rubric may reject a biologically correct result because it uses a different abundance measure or a different taxonomic identifier. For example, one database may report *Bottlenose Dolphin Adenovirus 1*, while another reports the synonymous abbreviation *BdAdV-1*.

In this case, the natural response is to introduce an interpretive grader, which in practice amounts to using an LLM judge due to scale considerations. This introduces additional variability, not only due to the nature of LLMs themselves, but due to the fact that the scoring now becomes a function of additional specification of how the LLM grader should apply the rubric, i.e. we introduce an implicit *strictness criterion*, which is another parameter that can be tuned. Different judges, whether human or LLM-based, may apply the same criteria differently, and an identical LLM judge may also produce different judgments for the same input samples.

BiomniBench (Qu et al., 2026) compared five candidate LLM graders against expert A/B/C labels for 597 individual rubric criteria drawn from 35 tasks. Gemini 3.1 Pro, which was subsequently used for the paper's reported experiments, achieved the highest agreement with the expert labels: 82% exact agreement and a linear-weighted Cohen's κ of 0.70. Thus, even the best-performing judge assigned a different rating from the expert on approximately one in five criteria.

One of the ways of mitigating these issues, at some cost to biological realism, is to replace the empirical reference with a simulated target. Here, the benchmark designer controls the data-generating process by, for example, simulating data to inject a known signal so that the target is known by construction and can be graded deterministically. CompBioBench (Nair et al., 2026) uses this strategy to create tasks such as identifying a known contaminating species. GeneBench-Pro (Li & Ho, 2026) goes further by simulating the full data-generating process.

Constructed targets reduce ambiguity in both the reference answer and its grading. This control comes at the cost of *construction bias*. The benchmark is valid only within the world created by its designers, and the injected signals may be easier to detect than their natural counterparts. GeneBench-Pro (Li & Ho, 2026) mitigates this risk by testing alternative workflows to ensure that they produce distinguishable answers.

At the grading stage, judge effects comprise both a systematic and a variable component:

$$J_{\text{effects}} = B_{\text{judge}} + V_{\text{judge}}, \tag{1}$$

The reported evaluation error for an agent f can be understood as accumulating across several stages of the evaluation process. *Construction bias* arises from how the benchmark world and its target answers are created. *Specification bias* is introduced when those targets are translated into prompts,

rubrics, thresholds, etc. Even on a fixed dataset, repeated executions of the same agent may produce different trajectories, creating *execution variability*. Finally, applying the rubric introduces judge effects: systematic *judge bias* as well as variability across repeated judgments. Conceptually, these contributions can be summarized as

$$\mathcal{E}_{\text{eval}}(f) \approx B_{\text{construction}} + B_{\text{specification}}(f) + B_{\text{judge}}(f) + V_{\text{execution}}(f) + V_{\text{judge}}(f). \quad (2)$$

Equation 2 is not intended as a formal statistical identity. Rather, it identifies the mechanisms through which a reported score can diverge from the scientific capability that the benchmark was supposed to measure.

The rubric–grader matrix exposes errors introduced during benchmark construction, task and grading-contract specification, or application of that contract by a judge. At the final stage, systematic *judge bias* differs from *judge variance* across repeated judgments under otherwise identical conditions. These sources of evaluation error are summarized in Table 3.

Table 3: Sources of error in benchmark evaluation.

Component	**Description**	**Typical sources**
Construction bias	Introduced by the design of the evaluation target or the benchmark.	Simulated or injected signals, assumptions embedded in reference answers, task selection, and reduced biological realism.
Specification bias	Introduced when the evaluation target is translated into a formal task and grading contract.	Underspecified prompts, numerical thresholds, accepted vocabulary, output schemas, and anticipated outcome sets.
Judge bias	Introduced when a human or LLM judge (the system prompt) applies the grading specification.	Consistent strictness or leniency, stable differences between judges, interpretations of rubric criteria.
Judge variance	Introduced when the same grading specification is applied repeatedly under otherwise equivalent conditions.	Sampling variability, model nondeterminism, and inconsistent judgments across repeated evaluations by the same judge.

Evaluation awareness and benchmark maximization. An additional difficulty comes up when an agent recognizes that it is being evaluated and conditions its behavior on the expected grader rather than the scientific task. Frontier models can distinguish evaluation transcripts from deployment interactions substantially better than chance, particularly in agentic settings (Needham et al., 2025). Banerjee (2026) found that Kimi-K3 reportedly referred to the grader, reference solution, or evaluation construction in approximately 61% of 2,544 trajectories, sometimes selecting methods or modifying outputs to match an inferred answer key. The resulting score may therefore reflect an agent's ability to optimize for the evaluation process rather than its underlying scientific capability, further complicating our work.

4 Agent failure modes, how we diagnose them, and what they tell us

The main goal of the evaluation is to learn the capabilities of the AI systems and if we can trust the process by which they arrive at results in computational biology. In essence, the goal is to determine how the jagged intelligence concept (Dell'Acqua et al., 2023) is manifested in agents for biological data analysis. *Jagged intelligence* describes how AI systems, compared to a human reference, can excel at complex tasks while unexpectedly failing at seemingly simple ones, making their capabilities difficult to predict (Karpathy, 2024). To do that, we need to look beyond the benchmark scores and examine where, how, and why agents fail.

To that end, Table 4 sorts the failures into several classes that can be probed differently:

Table 4: Single-trajectory failure modes of AI agents in biological data analysis.

Failure category	**Description**
General agentic failures	Failure to reliably plan, execute, monitor or perform error recovery
Data and grounding failures	The agent incorrectly selects, acquires, checks, represents, or understands biological data or external knowledge sources.
Analysis design and workflow failures	The agent selects an analysis strategy or workflow that does not adequately address the scientific question.
Statistical inference failures	The quantitative analysis uses wrong methods or parameters, the uncertainty is represented inadequately, and results don't correspond to the analysis
Interpretation failures	The agent assigns biological or scientific meaning that is not justified by the data and analysis.

4.1 General agentic failures

Evaluating general agentic competence is the necessary starting point for evaluation. These failures include early stopping, reasoning stuck in loops, inability to execute tools and software correctly, perform error correction, manipulate data formats, etc. Currently reported benchmark performance shows that frontier agents have the general competence required for running biological data analysis end-to-end. We therefore turn our attention to currently more relevant modes of failure. [3].

4.2 Data and grounding failures

Data and grounding failures are the next place we look into once execution succeeds. This is where a generic coding agent is supposed to show the first signs of biological reasoning. Even though certain agents may achieve high completion rates and arrive at the correct result as intended by designers, how certain are we that under slightly different conditions we would reach the same result? Luckily for us, this is the failure mode that has the clearest pathway for testing it.

Adversarial perturbation tests can help us analyze this failure mode. BioAgent Bench (Fa et al., 2026) introduces a way of quantifying this behavior by introducing targeted tests for data and grounding competence, which become a part of the evaluation suite:

- **a) data corruption** – the agent is supposed to recognize the unnatural distribution of some quantity in the input data
- **b) decoy data** – the agent is provided with a data file of an organism completely unrelated to the original task

What the analysis of the trajectory finds is that agents commonly continue on with the analysis despite finding something wrong with the data, or indiscriminately include all of the supplied organisms (even the wrong ones) with the same file extension.

In SpatialBench-Long (Diks et al., 2026a), 12 of 75 selected trajectories were categorized as missing important metadata or sanity-checks, with deliberately swapped condition labels providing the clearest cases of data grounding failures. Liu et al. (2026) report analogous failures, including using the wrong cell-label field and failing to harmonize gene identifiers before matching single-cell and spatial data.

Such failures reveal that the agent has both a prior and a measurement, but it does not reliably decide when the supplied evidence should defeat the prior. This is more diagnostic of biological reasoning than a simple tool failure because the code can run perfectly while the biological conclusion is wrong.

[3]The contribution of harnesses or the tooling around models vs. knowledge inherent to the model itself we leave to individual benchmarks

4.3 Analysis design and workflow failures

We define this failure mode as choosing an analysis that does not answer the intended question, or, on the computational side, misapplication of computational biology workflows and algorithms.

EpiBench (Muralidharan et al., 2026) agents used end-to-end rather than local alignment for CUT&RUN spike-in reads, counted the two strand-specific Bismark rows for one CpG as independent sites, or passed paired-end BAMs directly to MACS3 instead of constructing the intended Tn5 insertion-site representation. These choices changed normalization, methylation effect sizes, peak calls, and downstream motif results.

Although we don't see it in practice yet, because of the computational burden, benchmark designers could evaluate analysis design first by evaluating the trajectories across multiple identical runs through monitoring the robustness of the chosen analysis path and the parameter selection. Secondly, a more detailed diagnostic procedure could estimate how robust the analysis path is under different samples and task prompts. A potential targeted test that could simplify the analysis of this failure mode could be made by inserting *diagnostic triggers*, in the form of data or prompts that should initiate an appropriate change of method.

4.4 Statistical reasoning failures

Statistical inference and interpretation failures are hardest to measure because they're grounded in some empirical data. These failures occur when an agent chooses the wrong estimand, experimental unit, model, uncertainty statement, etc. This is an important failure mode because syntactically correct code and a small p-value can support an invalid claim.

GeneBench-Pro (Li & Ho, 2026) shows multiple examples of this failure mode and describes it as *notice–act gap*, where agents identify an irregularity but fail to propagate its consequences downstream. For example, in a pharmacogenomic survival task, an otherwise reasonable time-varying Cox model failed to address treatment–confounder feedback.

This failure mode is minimized by using constructed targets, because the data-generating process and target estimand can be controlled, allowing for interventions on confounding while holding the scientific question fixed. Tasks with simulated targets should then be paired with empirical tasks, as we see in GeneBench-Pro (Li & Ho, 2026), to reduce the effect of construction bias.

4.5 Interpretation failures

Interpretation failures are where the distinction between biological knowledge and biological reasoning becomes clearest. This failure is apparent when a result is translated into a biological claim whose identity, importance, direction, or causal strength is not supported by the analysis. scBench-Long (Diks et al., 2026b) supplies several controlled examples. In melanoma tasks, models favored familiar markers such as CD39 and PD-1 even though the requested definition of tumor reactivity required physical tumor/APC engagement together with paired TCR clonal expansion.

EpiBench (Muralidharan et al., 2026) found seven reviewed evaluations in which GPT-5.5 computed the correct result but did not submit it, while in four, the result was visible in the trajectory, but was replaced by a more familiar biological expectation. HeurekaBench (Panigrahi et al., 2026) similarly observed agents substituting the mean expression of a few recalled canonical markers for a pathway-enrichment analysis or using literature knowledge to eliminate multiple-choice options rather than deriving the answer from the dataset.

Interpretation failures are the most diagnostic in evaluating biological reasoning. All the evidence has been compiled and laid bare before the agent, and the workflow was done correctly. Still, even in that ideal case, we can't trust the agent to reliably choose between the inherent familiar knowledge and the evidence derived from the data.

5 Conclusion

We began by asking whether current agent evaluations can distinguish successful workflow execution from trustworthy scientific judgment. Current evaluations find fragments of this capability, but not

yet the consistency required for trust. Across current bioinformatics evaluations, frontier agents show genuine but jagged competence. They reliably execute pipelines. However, deeper probing reveals that pipeline success is not followed by reliable scientific judgment.

To determine the point when agents are reliable enough to go from supervised collaborators to autonomous research systems, benchmarks should report whether the agent's reasoning and conclusions respond appropriately when the evidence changes, along with the outcome grading. Some recent benchmarks have begun to probe this distinction through perturbation tests, trajectory diagnostics, and tasks pairing empirical targets with counterfactual or simulated alternatives. These tests should now be automated, standardized, and reported as first-class metrics alongside outcome grades.

Until agents can consistently pass such tests, claims of autonomy will certainly mistake successful workflow execution for trustworthy scientific reasoning.

AI USE STATEMENT

Generative AI assisted with writing and literature retrieval; the authors verified all claims, analyses, and interpretations, made the final editorial decisions, and take responsibility for the paper.

REFERENCES

Samuel Alber, Bowen Chen, Eric Sun, Alina Isakova, Aaron J. Wilk, and James Zou. Cellvoyager: Ai compbio agent generates new insights by autonomously analyzing biological data. *Nature Methods*, 23(4):749–759, March 2026. ISSN 1548-7105. doi: 10.1038/s41592-026-03029-6. URL http://dx.doi.org/10.1038/s41592-026-03029-6.

Anthropic. Demystifying evals for ai agents. Anthropic Engineering, January 2026. URL https://www.anthropic.com/engineering/demystifying-evals-for-ai-agents.

Arjun Banerjee. Surfacing benchmark-maxxing in Kimi-K3. LatchBio, July 2026. URL https://blog.latch.bio/p/surfacing-benchmark-maxxing-in-kimi. Accessed: 2026-08-14.

Aashka Bhowmick, Nasim Rahmatpour, Shivang Sharma, Qian Xu, Ananya Gupta, Asmita Lagwankar, Arjun Banerjee, and Christopher Zou. Variantbench: An agentic benchmark for genetic variant discovery and interpretation, 2026. URL https://latch.bio/variantbench. Technical report, LatchBio.

Dechao Bu, Jingbo Sun, Kun Li, Zihao He, Wei Huang, Jinlin Hu, Shanshan Zhang, Shuangshuang Lei, Peipei Huo, Zhihao Wang, Sheng Wang, Tao Wang, Kai Gao, Yang Wu, Lianhe Zhao, Kai Wang, Gen Li, Huan Song, Yang Jin, Kang Zhang, Runsheng Chen, and Yi Zhao. Empowering ai data scientists using a multi-agent llm framework with self-evolving capabilities for autonomous, tool-aware biomedical data analyses. *Nature Biomedical Engineering*, March 2026. ISSN 2157-846X. doi: 10.1038/s41551-026-01634-6. URL http://dx.doi.org/10.1038/s41551-026-01634-6.

Fabrizio Dell'Acqua, Edward McFowland III, Ethan R. Mollick, Hila Lifshitz-Assaf, Katherine Kellogg, Saran Rajendran, Lisa Krayer, François Candelon, and Karim R. Lakhani. Navigating the jagged technological frontier: Field experimental evidence of the effects of artificial intelligence on knowledge worker productivity and quality. Technology & Operations Management Unit Working Paper 24-013, Harvard Business School, 2023. URL https://ssrn.com/abstract=4573321.

Ian Diks, Harihara Muralidharan, Tim Proctor, and Kenny Workman. Verifiable benchmarking of long-horizon spatial biology. *arXiv preprint arXiv:2605.28065*, 2026a.

Ian Diks, Zhen Yang, Arjun Banerjee, Tim Proctor, and Kenny Workman. scbench-long: Verifiable benchmarking of long-horizon single-cell biology. *arXiv preprint arXiv:2606.26563*, 2026b.

Dionizije Fa and Marko Culjak. Sound agentic science requires adversarial experiments. *arXiv preprint arXiv:2604.22080*, 2026. doi: 10.48550/arXiv.2604.22080. Published at the ICLR 2026 Workshop on Agents in the Wild.

Dionizije Fa, Marko Culjak, Bruno Pandza, and Mateo Cupic. Bioagent bench: An ai agent evaluation suite for bioinformatics. *arXiv preprint arXiv:2601.21800*, 2026. doi: 10.48550/arXiv.2601.21800. ICML 2026.

Spandan Garg, Benjamin Steenhoek, and Yufan Huang. Saving swe-bench: A benchmark mutation approach for realistic agent evaluation, 2026. URL https://arxiv.org/abs/2510.08996.

Ali E. Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, Caralyn J. Szostkiewicz, Dmytro Shved, Gavin J. Gyimesi, Jon M. Laurent, Samantha M. Wright, Muhammed T. Razzak, Andrew D. White, Silvia C. Finnemann, Michaela M. Hinks, and Samuel G. Rodriques. A multi-agent system for automating scientific discovery. *Nature*, 655(8122):497–505, May 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10652-y. URL http://dx.doi.org/10.1038/s41586-026-10652-y.

Andrej Karpathy. Jagged intelligence. X (formerly Twitter), July 2024. URL https://x.com/karpathy/status/1816531576228053133. Post, July 25, 2024.

Jeremy Li and Andrew Ho. Genebench-pro: Evaluating multistage statistical reasoning in genomics, quantitative biology, and translational biomedicine. *bioRxiv*, 2026. doi: 10.64898/2026.06.29.735386. URL https://www.biorxiv.org/content/10.64898/2026.06.29.735386v1.

Yang Liu, Lu Zhou, Xiawei Du, Ruikun He, Xuguang Zhang, Rongbo Shen, and Yixue Li. Benchmarking LLM-based agents for single-cell omics analysis. *Genome Biology*, 27(1):123, 2026. doi: 10.1186/s13059-026-03998-z. URL https://doi.org/10.1186/s13059-026-03998-z.

Ludovico Mitchener, Jon M. Laurent, Alex Andonian, Benjamin Tenmann, Siddharth Narayanan, Geemi P. Wellawatte, Andrew White, Lorenzo Sani, and Samuel G. Rodriques. BixBench: A comprehensive benchmark for LLM-based agents in computational biology. *arXiv preprint arXiv:2503.00096*, 2025. doi: 10.48550/arXiv.2503.00096. URL https://arxiv.org/abs/2503.00096.

Harihara Muralidharan, Reema Baskar, Soo Hee Lee, Tim Proctor, and Kenny Workman. Epibench: Verifiable evaluation of ai agents on epigenomics analysis. *arXiv preprint arXiv:2606.13602*, 2026.

Surag Nair, Laura Gunsalus, Brian Orcutt-Jahns, Jordan Rossen, Avantika Lal, Carlo De Donno, Muhammed Hasan Celik, Kipper Fletez-Brant, Xiaoman Xie, Hector Corrada Bravo, and Gokcen Eraslan. Agentic systems are adept at solving well-scoped, verifiable problems in computational biology. *bioRxiv*, 2026. doi: 10.64898/2026.04.06.716850. URL https://www.biorxiv.org/content/10.64898/2026.04.06.716850v1.

Joe Needham, Giles Edkins, Govind Pimpale, Henning Bartsch, and Marius Hobbhahn. Large language models often know when they are being evaluated. *arXiv preprint arXiv:2505.23836*, 2025. doi: 10.48550/arXiv.2505.23836. URL https://arxiv.org/abs/2505.23836.

Siba Smarak Panigrahi, Jovana Videnović, and Maria Brbić. Heurekabench: A benchmarking framework for AI co-scientist. *arXiv preprint arXiv:2601.01678*, 2026. doi: 10.48550/arXiv.2601.01678. URL https://arxiv.org/abs/2601.01678.

Yibo Qiu, Zan Huang, Zhiyu Wang, Handi Liu, Yiling Qiao, Yifeng Hu, Shu'ang Sun, Hangke Peng, Ronald X Xu, and Mingzhai Sun. Biomars: A multi-agent robotic system for autonomous biological experiments, 2025. URL https://arxiv.org/abs/2507.01485.

Yuanhao Qu, Kaixuan Huang, Ming Yin, Kanghong Zhan, Dyllan Liu, Di Yin, Henry C. Cousins, William A. Johnson, Xiaotong Wang, Mihir Shah, Russ B. Altman, Denny Zhou, Mengdi Wang, and Le Cong. Crispr-gpt for agentic automation of gene-editing experiments. *Nature Biomedical Engineering*, 10(2):245–258, 2025. ISSN 2157-846X. doi: 10.1038/s41551-025-01463-z. URL http://dx.doi.org/10.1038/s41551-025-01463-z.

Yuanhao Qu, Yingzhou Lu, Xinming Tu, Serena Zhang, Tianwei She, Alexander Glenn Shaw, Jou-Ho Shih, Bingqing Zhao, Minjie Shen, Haochen Yang, Jielin Yan, Rongchuan Zhang, Xinze Wu, Tingting Li, Le Cong, Xiaobo Hu, Yuan Jiang, Jiayun Dong, Tao Peng, Jure Leskovec, and Kexin Huang. Biomnibench: Process-level evaluation of LLM agents for real-world biomedical research. *bioRxiv*, 2026. doi: 10.64898/2026.05.12.724604. URL https://www.biorxiv.org/content/10.64898/2026.05.12.724604v1.

Qiushi Sun, Zhoumianze Liu, Chang Ma, Zichen Ding, Fangzhi Xu, Zhangyue Yin, Haiteng Zhao, Zhenyu Wu, Kanzhi Cheng, Zhaoyang Liu, et al. Scienceboard: Evaluating multimodal autonomous agents in realistic scientific workflows. *arXiv preprint arXiv:2505.19897*, 2025. doi: 10.48550/arXiv.2505.19897. URL https://arxiv.org/abs/2505.19897. ICLR 2026.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

Kenny Workman. Human verification of spatialbench, May 2026. URL https://blog.latch.bio/p/human-verification-of-spatialbench. Blog post, accessed 2026-08-13.

Kenny Workman, Zhen Yang, Harihara Muralidharan, Aidan Abdulali, and Hannah Le. scbench: Evaluating ai agents on single-cell rna-seq analysis. *arXiv preprint arXiv:2602.09063*, 2026a.

Kenny Workman, Zhen Yang, Harihara Muralidharan, and Hannah Le. Spatialbench: Can agents analyze real-world spatial biology data? *arXiv preprint arXiv:2512.21907*, 2026b.