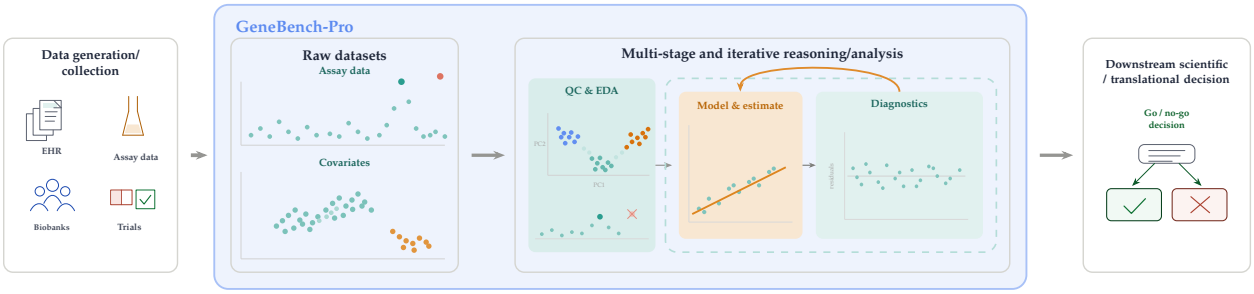
GeneBench-Pro: Evaluating Multistage Statistical Reasoning in Genomics, Quantitative Biology, and Translational Biomedicine

Jeremy Li[†], Andrew Ho[†]
OpenAI

June 30, 2026



Abstract

We introduce GeneBench-Pro, an expanded and improved version of GeneBench that comprises harder problems across a wider breadth of domains. GeneBench-Pro is a benchmark for AI agents performing realistic multi-stage scientific analyses in genomics, quantitative biology, and translational biomedicine which seeks to capture the complexity of real-world problems that computational life scientists face when tasked with producing a conclusion upon which a downstream scientific or translational decision is contingent. The benchmark comprises 129 evaluations targeting quantities of direct practical relevance across 10 primary domains and 21 terminal subdomains, with a genomics-centered core. Similarly to GeneBench, each problem provides the agent with brief context, a target estimand, and minimal guidance otherwise; the agent must then navigate multiple dependent decision points; *i.e.*, substantive inferential forks where a plausible wrong choice changes the downstream analysis, to identify and execute the correct analysis workflow and arrive at the correct answer. Relative to GeneBench, GeneBench-Pro adds 29 new problems, drops three, and introduces significantly redesigned versions of 54 of the remaining 100 overlapping problems. 82 of the 129 problems were reviewed by external domain experts, whose findings led to prompt/data modifications and redesign of those problems whose targets were not sufficiently identifiable. Ten externally reviewed problems are released publicly, 50 held-out problems were provided to Artificial Analysis for independent third-party model benchmarking, and the remainder are retained as an internal holdout. In evaluations over the full 129-problem suite, GPT-5.6 Sol reaches an eval-level pass rate of 28.7% at the `max` reasoning level, and GPT-5.6 Sol Pro reaches 31.5% in separately reported GPT Pro runs. GPT-5.5 reaches 12.0%, GPT-5.4 reaches 8.9%, and the strongest non-GPT baseline, Claude Opus 4.8, reaches 16.0%. As with GeneBench, models often complete substantial portions of the workflow but exhibit a consistent gap between *noticing* and *acting* by identifying local diagnostic signals but failing to propagate the implications to the corresponding analysis decision. As a result, models often select wrong estimators or persist on initially plausible but incorrect analysis paths. GeneBench-Pro therefore measures an emerging capability of long-horizon biological reasoning that remains unreliable.

[†]Corresponding authors: `jeremiah.li@c-openai.com`; `ajh@openai.com`.

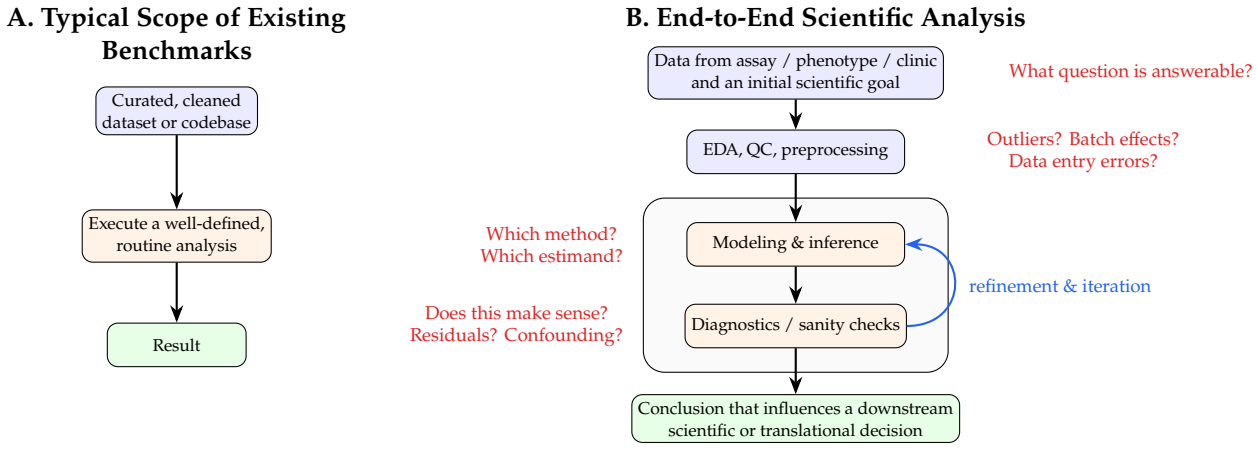


Figure 1: The benchmark gap. (A): many existing biology AI benchmarks begin from a curated dataset and a highly specified prompt and evaluate a narrowly scoped analysis step with a cleanly verifiable answer. **(B)**: real-world scientific analysis more often spans a wider and more iterative process: data of unknown quality are obtained from some external source, and analysts must decide what questions are answerable, perform quality control and exploratory analysis, choose models and estimands, diagnose failures, and ultimately reach a conclusion that can influence the next scientific or translational decision. GeneBench-Pro is intended to evaluate this broader process rather than just its constituent components.

Introduction

Agentic systems now perform strongly on basic software engineering evaluations, while newer suites such as SWE-Bench Pro, SWE-Lancer, DeepSWE, and FrontierSWE have shifted measurement toward longer-horizon, economically grounded, and deeper repository-level work;[1–4] broader evaluation efforts such as FrontierScience, FrontierMath, and Humanity's Last Exam target difficult expert-level and novel-problem settings,[5–7] and METR's recent time-horizon analyses likewise suggest that the duration of tasks frontier agents can complete autonomously is increasing rapidly.[8] Simultaneously, biology foundation models such as ESM3, Evo 2, and Omnii have pushed protein and genome modeling to new scale and fidelity.[9–11]

Yet there has been relatively little formal examination of AI performance on the broader routine process that underpins much of modern life science research: executing a multi-step quantitative analysis starting from potentially errorful raw data, proceeding through a series of contingent procedures requiring statistical reasoning and strategic judgement, and ending at a decision-relevant conclusion (see **Figure 1B** for a high-level schematic of the typical process). This class of work is a major practical bottleneck in data-rich fields including genomics, proteomics, transcriptomics, and metabolomics; for example, recent reviews in the genomics literature argue that as sequencing has scaled, downstream computation and analysis, rather than data generation, have become the central bottleneck.[12–14]

In contrast to most engineering tasks, scientific research is far more iterative, open-ended, and ambiguous. Its core challenges stem not from the execution of analytical workflows, but from the importance of scientific intuition or "research taste": chains of judgment calls about what question the data can support, what data to include, which estimand or model is appropriate, whether diagnostics invalidate initial hypotheses, and when the evidence is strong enough to support a conclusion.

Older biology benchmarks mostly covered narrower forms of the workflow or emphasize breadth over depth (see **Figure 1A**), while newer efforts have begun moving closer to longer-horizon and agentic biology work.[15–24] Yet these still comprise fairly narrowly scoped problems that have been devised with respect to specific real-world datasets — problems which, even worse, often are not convincingly shown to only have one unique answer rather than a *distribution* of possible valid answers resulting from one of many possible

defensible approaches, leading to difficulties in grading and uncertainty in whether ostensible failures should be taken at face value.

There therefore remains a gap in the literature for robust benchmarks which test whether agents can estimate quantities of biological interest from multi-stage analyses in which the accuracy of the final estimate is contingent upon accurate upstream statistical reasoning and diagnostic decision-making. GeneBench[25] was our first effort to address this gap directly, and leveraged simulated data, which enabled us to design sophisticated problems requiring multiple steps of contingent statistical decision-making while remaining confident in the fidelity of model grading.

Here we introduce GeneBench-Pro, an updated version of GeneBench comprising problems spanning industry and academically relevant domains that extends beyond the original scope of genomics into molecular and quantitative biology, pharmacogenomics, cancer biology, microbial genomics, clinical translation, and other settings where multistage statistical reasoning is required. As with GeneBench, each problem is a self-contained, multi-step analysis that provides (1) a realistic, messy dataset intended to reflect the data a scientist would receive from a lab, EHR system, or other collection pipeline, and (2) a *minimum viable prompt* which provides brief experimental context and defines a target estimand. That estimand is chosen to reflect a quantity that would inform a downstream decision in practice.

The current suite contains 129 evaluations across 10 primary domains and 21 terminal subdomains, with a genetics-centered core in population genetics, statistical genetics, quantitative genetics, and regulatory/molecular omics, and adjacent coverage in clinical genetics and pharmacogenomics, cancer somatic genomics and liquid biopsy, functional perturbation, proteomics, microbial genomics, and forensic genetics. GeneBench-Pro also adds a step of external scientific review, where the full design process, relevant files, and analytical setup for a given problem were exposed to an external domain expert to verify realism, scientific validity, and to solicit ideas on how to further harden the problem. This review process was performed for 82 of the 129 problems.

In evaluations over the full 129-problem suite, GPT-5.6 Sol reaches a 28.7% pass rate with `max` reasoning. At the highest performing reasoning level for each mainline GPT-family row, pass rate rises from 4.9% for GPT-5.2 to 8.9% for GPT-5.4, 12.0% for GPT-5.5, 16.5% for GPT-5.6 Luna, 23.3% for GPT-5.6 Terra, and 28.7% for GPT-5.6 Sol. Separately reported GPT Pro runs reach 8.5%, 16.3%, 20.5%, 23.6%, 28.5%, and 31.5% for GPT-5.2 Pro, GPT-5.4 Pro, GPT-5.5 Pro, GPT-5.6 Luna Pro, GPT-5.6 Terra Pro, and GPT-5.6 Sol Pro, respectively. Among the evaluated non-GPT models, pass rates range from 0.6% to 16.0%, with Claude Opus 4.8 the strongest non-GPT baseline. Manual examination of model-reported reasoning suggests that the main qualitative improvement in stronger models lies less in noticing the relevant diagnostic clues than in turning those observations into concrete corrective and model-selection decisions that move the analysis onto the correct path.

We first describe the scope of GeneBench-Pro using a high-level atlas of the problem space, introduce the main design constraints required to make this class of decision-heavy scientific analysis benchmarkable, and summarize the external-review process used to ensure that our problems are robust. We include an illustrative clinical genomics problem to make these design constraints concrete, then present benchmark-wide results and discuss qualitative improvements in model performance.

Of the 129 total GeneBench-Pro problems, we chose a subset of ten to open source. Another disjoint subset of 50 problems was chosen to serve as an external benchmark against which third parties may evaluate their own models via Artificial Analysis. The remainder remain an internal hold-out.

Benchmark Scope and Construction

GeneBench-Pro, a collection of 129 problems across 10 primary domains and 21 subdomains, measures whether an agent can identify and execute the quantitative analysis required to estimate the target estimand from potentially errorful datasets with minimal guidance. **Figure 2** illustrates the domain coverage of the current suite.

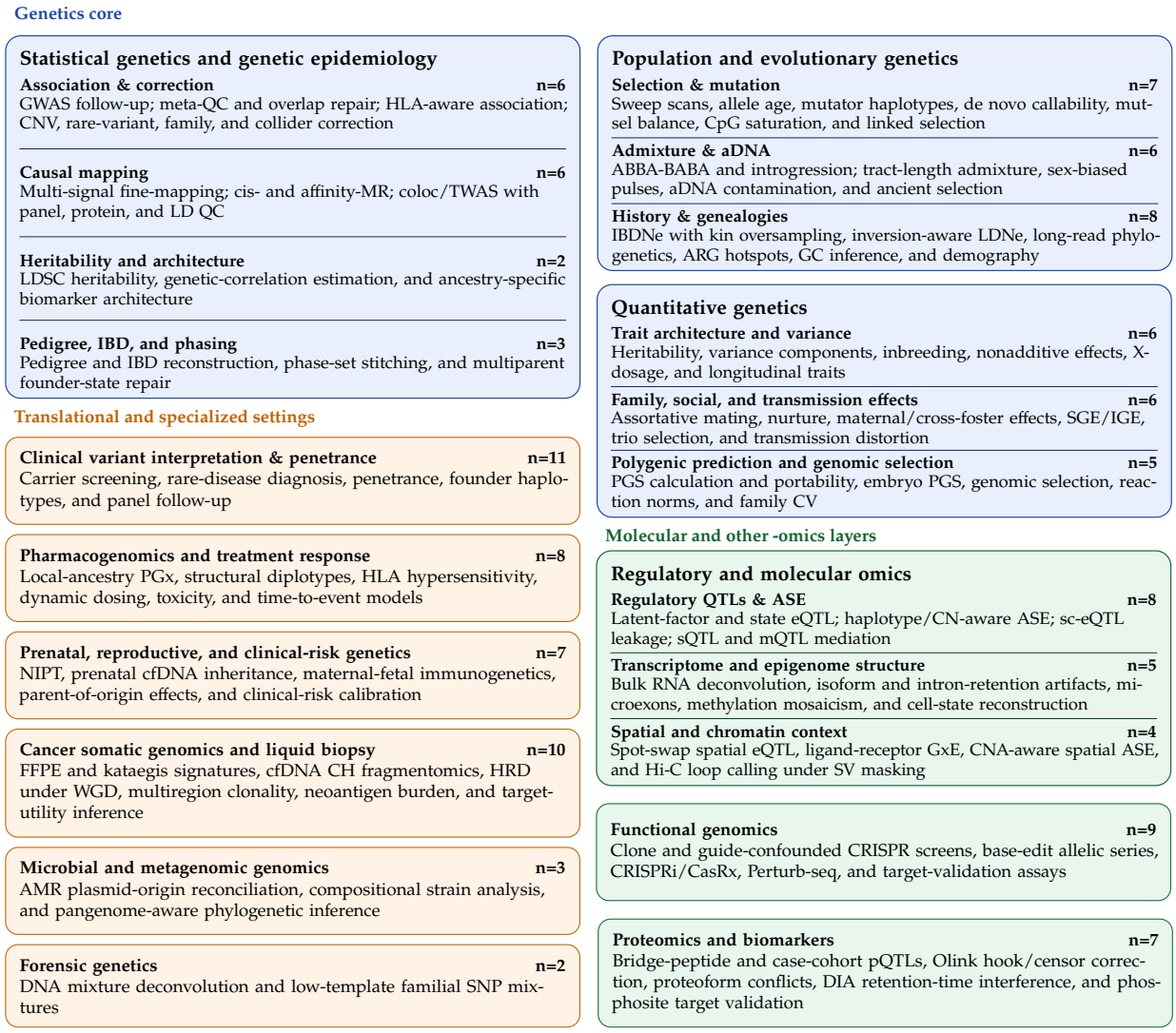


Figure 2: Domain atlas of the current GeneBench-Pro suite. GeneBench-Pro comprises 129 problems across 10 primary taxonomy domains and 21 terminal subdomains. Nested subcards expose the terminal subdomains within the larger domains.

Abbreviations: GWAS, genome-wide association study; QC, quality control; HLA, human leukocyte antigen; CN, copy number; CNV, copy-number variant; SV, structural variant; MR, Mendelian randomization; coloc, colocalization; TWAS, transcriptome-wide association study; LD, linkage disequilibrium; LDSC, linkage disequilibrium score regression; IBD, identity by descent; PGx, pharmacogenomics; CpG, cytosine-phosphate-guanine; aDNA, ancient DNA; ABBA-BABA, four-taxon introgression test; IBDNe and LDNe, effective population-size inference from identity-by-descent and linkage disequilibrium, respectively; ARG, ancestral recombination graph; GC, gene conversion; PGS, polygenic score; SGE, social genetic effects; IGE, indirect genetic effects; QTL, quantitative trait locus; ASE, allele-specific expression; eQTL, expression quantitative trait locus; pQTL, protein quantitative trait locus; mQTL, methylation quantitative trait locus; sQTL, splicing quantitative trait locus; sc-eQTL, single-cell expression quantitative trait locus; CRISPR, clustered regularly interspaced short palindromic repeats; CRISPRi, CRISPR interference; CasRx, an RNA-targeting CRISPR effector; Hi-C, genome-wide chromosome conformation capture; GxE, gene-by-environment interaction; CNA, copy-number alteration; NIPT, noninvasive prenatal testing; cfDNA, cell-free DNA; FFPE, formalin-fixed, paraffin-embedded; CH, clonal hematopoiesis; HRD, homologous recombination deficiency; WGD, whole-genome doubling; AMR, antimicrobial resistance; SNP, single-nucleotide polymorphism; DIA, data-independent acquisition.

Across the benchmark, an agent must filter and correct data, identify QC or ascertainment problems, choose methods, perform statistical inference, revise the analysis when intermediate results disagree with the initial plan/hypothesis, and produce a final quantitative answer. Many problems are framed as decision points in genetics-backed drug discovery and translational research, such as whether a GWAS signal survives correction strongly enough to advance and which gene or protein should be nominated as the likely effector target, while others are framed around more academically oriented questions, such as whether an observed pattern is better explained by selection or demography and which pedigree, haplotype, or ancestry reconstruction is supported by the data.

GeneBench-Pro preserves the same problem setup as GeneBench—a minimum viable prompt, agent-visible staged data files, and a defined output schema, but enhances the difficulty of the problems, expands the problem set, and shifts more of the difficulty away from addressing messy data toward decision-heavy statistical reasoning. Specifically, we began with GeneBench's 103 problems—we then withdrew three due to fatal issues identified; of the remaining original hundred, we significantly redesigned and hardened 54; we then added 29 new problems, resulting in a total of 129 problems.

Relative to GeneBench, GeneBench-Pro also increases coverage in clinical and translational genomics, pharmacogenomics, liquid biopsy, proteomics, microbial genomics, and specialized molecular-omics settings, and includes extensive external scientific review of 82 problems in the full suite (described below).

Public Case Studies

For each of the ten public GeneBench-Pro packages, we provide, in addition to the data and prompt for each problem, detailed reports including a description of the data generating process, an analysis walkthrough corresponding with the correct answer, validation and ablation evidence, and general construction rationale. These are distributed with the Hugging Face dataset.

Benchmark Setup

Each GeneBench-Pro problem is packaged as a self-contained scientific analysis. The agent receives an isolated workspace containing a *minimum viable prompt*, staged files, and a standard scientific Python stack. The prompt specifies the scientific question/task and target estimand without explicitly prescribing the workflow to be executed. The files are intended to resemble what an analyst might actually receive from assays or clinical systems rather than cleaned toy datasets. Each problem involves a chain of dependent decision points such that an incorrect choice at any stage propagates into downstream errors and ultimately failure to recover the final correct target.

The agent operates in a realistic sandbox with the staged files, access to general-purpose scientific libraries including `numpy`, `pandas`, `scipy`, `scikit-learn`, `statsmodels`, `lifelines`, `matplotlib`, and `seaborn`, and standard genomics bioinformatics tooling such as PLINK 2.0, `pysnptools`, `bed-reader`, `bedtools`, `pysam`, etc. (see **Methods**)—though the problems are designed not to require or rely on access to these tools. **Supplementary Figure 1** shows a schematic of the agent environment. Success therefore depends both on the agent recovering the analysis from the data as well as accurate implementation of the relevant methods.

Principle	Benchmark requirement	Failure mode if violated
Ground truth and identifiability		
Recoverable target	Agents are graded on recovering the quantity that is actually recoverable from agent-visible data, and not the hidden data-generating parameters.	Correct analyses can be marked wrong because the parameter under which data were generated is unrecoverable (*e.g.*, due to sampling variation in the DGP).
Unique, identifiable answer	The staged evidence along with a minimum viable prompt supports one uniquely defensible answer. If multiple approaches would naively appear reasonable, the data or prompt contain some empirical constraint that reasonably rules out all but one.	The task becomes under-specified, and success depends more on workflow preference rather than reasoning from the available evidence.
Clear numerical separation from incorrect answers	A comprehensive ablation suite demonstrates that plausible but incorrect analyses and shortcut methods yield materially different answers and fail by clear margins.	Wrong analyses can be graded as correct.
Problem specification		
Minimal viable prompt	The scientific question and graded estimand(s) are clearly defined, but the prompt does not give prescriptive instructions on how to perform the analysis.	The task either turns into evaluating how well agents can execute a well-defined analysis, or leaves multiple defensible interpretations of how the final answer should be estimated.
Robust QC requirements	When QC is required as part of reaching the solution, nearby reasonable thresholds lead to the same graded outcome.	The benchmark measures arbitrary cutoff choice rather than recognition of the qualitative problem.
Scientific workflow fidelity		
Fully simulated problems	Simulating data allows us to tune each aspect of the data-generating process such that realism, multi-stage inference, effect sizes, and diagnostic clues can be precisely controlled.	Difficulty, answer separation, and correctness may become difficult to calibrate.
Multi-stage inference	Problems require navigating ordered inferential forks in which upstream filtering, representation, and statistical model choices materially affect the final graded endpoint.	The benchmark reflects smaller, self-contained units of end-to-end analysis rather than the full scientific workflow.
Representation and nuisance variable discovery	Problems require recovery of the relevant scale, orientation, harmonization state, selection or censoring process, batch or latent-group structure, and QC-valid sample set before final estimation.	A statistically valid final model is applied to the wrong data or population, on the wrong scale, or on the wrong conceptual level.
Ablation-supported workflow adjudication	Plausible expert workflows are evaluated through ablations and sensitivity analyses, separating reasonable alternatives from choices that alter an identifiable stage of the inferential chain.	Success depends on an unreported analyst preference rather than a recoverable scientific target.
Observed failures occur across multiple decision points	Trace analyses display agent failures across multiple decision points rather than clustering around only a single decision point.	The problem effectively evaluates how well agents navigate a single difficult step rather than multiple difficult steps.
Literature-defensible solution	The intended correct solution involves standard or otherwise well-supported methods.	Success depends on benchmark-specific ad-hoc methods or unsupported analysis choices rather than well-established methods.

Table 1: Primary design constraints in GeneBench-Pro. Together, these are intended to keep the graded endpoint scientifically identifiable while preserving realistic ambiguity and data messiness.

Construction, Validation, and Grading

Open-ended scientific analysis is difficult to benchmark precisely because real data often admit multiple defensible analysis choices. For example, QC thresholds, model parameterizations, and reporting conventions can vary across analysts without there being only a single unambiguously correct analytical choice. If the outcomes of a benchmark change because one agent uses one defensible cutoff or convention while another agent uses a different, yet equally defensible one, this might reflect the arbitrary nature of that benchmark's design choices rather than the quality of scientific reasoning.

This is a particularly underappreciated issue in many existing long-horizon biology benchmarks, which are often based on real historical datasets around which new, multi-step questions are devised. In such data, extreme ambiguity is often present, to the extent that if one is, for example, tasked to apply a three-step QC or modeling pipeline to some dataset, there is rarely necessarily exactly one correct path through this garden of forking paths — at each of these three steps, there is a substantial probability that there exists an equally defensible choice that was not considered by the benchmark designer that ultimately leads to agents which make this choice to be graded as a failure, leading to a naturally decaying terminal pass rate with the length of the required inferential chain just by virtue of real-world messiness.

This can then lead to silent failure modes in which a benchmark may appear difficult due to the contribution of this mode of "failure" to overall benchmark performance, making it difficult to evaluate exactly what it is that such benchmarks are measuring. A useful benchmark for this type of work must therefore be insensitive to nearby defensible analyst choices, but sensitive to missing scientifically necessary stages.

In order to implement these multi-stage ("cascaded") setups in a way where we can be confident that a "fail" actually represents a scientific failure and not a result of the aforementioned natural attrition, GeneBench-Pro problems are based on constructively simulated problems where the full causal structure is known and where we simulate the full data-generating process (DGP). Simulation allows us to directly tune the complexity and difficulty of this cascade while ensuring that (1) QC-sensitive decisions are robust to small researcher-choice variation, (2) plausible wrong analyses fail for substantive reasons, and (3) the graded endpoint is actually recoverable from the agent-visible data.

We quantify this cascaded structure through the number of *decision points* in each problem: substantive inferential forks where a plausible wrong choice leads to a qualitatively different downstream answer. The number of these decision points ranges from 3 to 13 across the current suite (with a median of 6).

Operationally, problem development begins from a real-world analysis pattern and a target estimand. These real-world analysis patterns are synthesized from the literature and domain expertise to reflect common, high-impact scientific questions and workflows, and are specifically chosen so they do not recapitulate well-known textbook examples or papers, so as to avoid the risk of benchmarking against memorized solutions. Data are then simulated so that the correct answer is recoverable from the staged files (for example, the maximum likelihood estimate of a parameter resulting from the correct approach would be considered as the ground truth value for grading, rather than the parameter under which the data were generated). A minimum viable prompt containing the minimum amount of information required to make the correct answer identifiable is then constructed. Prompts also include a standardized instruction that the data came from a real experiment, analytical reasoning quality is graded alongside numerical correctness, and the final response must be exactly one JSON object.

Once an initial draft of a problem is completed, extensive validation is performed. Results from analyses involving plausible but incorrect decisions at the various inference stages are checked via ablation and verified to be sufficiently distinct from the graded answer. Independent reviews for scientific validity, methodological soundness, and target identifiability are conducted in order to ensure that the evaluation is testing the intended capabilities rather than whether agents can infer an artifact-specific workflow preference that is not uniquely supported by the data. Problem drafts are then iteratively audited through multiple rounds of frontier-model pilots and detailed trace analyses in order to check for unintended leakage, alternative unintended pathways to the correct answer, prompt-grader mismatch, and robustness. This process is

intended to ensure that wrong-but-plausible analyses fail for substantive reasons and that passing runs reflect the intended inferential path rather than shortcuts. **Table 1** summarizes the main benchmark-level constraints that follow from these requirements.

GeneBench-Pro uses binary grading against recoverable targets under calibrated tolerances chosen to allow for numerical and implementation-level variation; the evaluation setup and package-level grading protocol is summarized in the **Methods**.

External Scientific Review

External review was used to assess a subset of candidate problems in order to gain broader confirmation of the scientific defensibility of the problems as well as their adherence to the stated benchmark design principles.

Reviews focused on target identifiability given the agent-visible prompt and data, method implementations, realism, and estimator choices. Feedback was adjudicated against the benchmark design constraints outlined in **Table** 1. This review layer also reinforced the benchmark's focus on statistical reasoning within quantitative biology: defining identifiable estimands, choosing among defensible finite-sample targets, checking method equivalence, respecting assay-specific artifacts, and recognizing when a realistic data-generating assumption changes the scientific answer.

Review Process

To limit exposure of unreleased tasks, reviewers received access to private repositories with their assigned problems. Each repository contained the prompt and data for each problem, a detailed problem report fully describing the problem design, validation materials, and an issue thread for their feedback.

The reviewer pool included 11 domain experts spanning graduate students, postdoctoral researchers, industry scientists, and professors. In total, 84 candidate problems received external review. 82 are included in the 129-problem evaluation, and two reviewed candidates were withdrawn due to fatal issues. The ten publicly released GeneBench-Pro problems were selected conditional on external review.

Review Themes and Resulting Changes

Table 2 gives representative excerpts from the reviews. For those problems where issues were identified via review, critiques tended to fall in a few categories: (1) the presence of hidden assumptions or ambiguity in estimand definitions which would result in non-identifiability of the graded target, (2) errors in the method implementations corresponding with the "correct" approach which would lead to correct answers being graded against an incorrect ground truth, (3) contrived setups which were nonrepresentative of real-world workflows.

Corrections were proportional to the finding: minor issues led to prompt clarifications or surgical changes to the data files, whereas more substantive issues led to more substantial work in modifying the DGP, ground-truth generation, and in extreme cases, complete redesign or withdrawal of the problem. After corrections were applied, the standard loop of agent pilots $\rightarrow$ trace examination $\rightarrow$ further revisions was repeated to generate the final candidate.

Finding type	Problem	Reviewer concern and context	Disposition or change
Underspecified estimand	Cis-MR with winner's curse and LD	"the exact graded solution rests on a particular pipeline not uniquely dictated by the prompt." The concern was that broad robust-MR, colocalization, or external-annotation strategies could be scientifically reasonable even if they did not match the specified ground truth pipeline.	Specified the residual-pruned finite-sample two-coefficient estimand in the prompt and answer contract.
Alternate defensible method failing	Recent-pulse sex-biased admixture	"I couldn't understand what the posterior represented" and whether a solver should "propagate uncertainty into the final solution." The reviewer identified a defensible posterior-weighted ancestry-fraction variant rather than a pure hard-call tract summary.	Allowed posterior-weighted ancestry fractions in grading, rather than requiring a hard-call result.
Method parity violation	Genetic correlation estimation via LDSC	The reviewer wrote that the ground truth was "not implementing actual LDSC" and that the critical issue was "wrong M, OLS not WLS." The concern was that the benchmark was grading an ad hoc LD-score estimator while presenting itself as an LDSC genetic-correlation task.	Fixed the reference implementation to implement LDSC properly.
Underspecified prompt	Somatic TXR1 target activation	"SV-mediated but that's not clear in the agent provided materials." Private materials made the causal SV mechanism clear, but the solver-facing materials left room for interpreting high copy number without an SV as activation.	Added "SV-driven" wording to the solver-facing target definition.
Ground truth not recoverable	Dynamic PGx treatment response	The reviewer recommended changing the "ground truth to the DGP's realized data" and noted that the validator was "suboptimal, a little ambiguous." Several calibration, toxicity, and biomarker nuisance specifications were defensible under the original prompt.	Aligned the ground truth with the realized simulated data and made the validator target and nuisance specifications explicit.
Non-identifiable phasing	Copy-number-aware phased ASE	The reviewer argued it was "not possible to have a fixed polarity" for arbitrary haplotype labels and that the flanking-marker step could not be used "in its current state." Marker relevance, copy-number context, and tumor purity were under-specified.	Redesigned the prompt and data with donor-specific phase blocks, explicit copy-number segments, marker relevance, and tumor-purity context.
Conflicting assay evidence	Long-read pseudogene phase diagnosis	"short reads and long reads" gave contradictory phasing evidence. The reviewer argued the target should state that phase posterior estimation should be based on long-read evidence, with QC used as pass/fail rather than as a continuous probability weight.	Specified long-read-only phase evidence and made phase-set quality a pass/fail QC screen.
DGP method implementation error	Assortative mating and indirect genetic effects	The reviewer found that the simulated child-parent PGI correlation "came out to be 0.38. This is too low" and traced it to a "faulty formula." The review also noted that the solver-visible heritability parameter should be h_f^2 rather than the originally exposed quantity.	Rebuilt the simulator around the corrected child/mother/father PGI covariance, exposed h_f^2, and updated the validator equations.

Table 2: Examples of review findings. Short quoted excerpts from external-review comments followed by brief context explaining the scientific concern are shown with the resulting changes that were made.

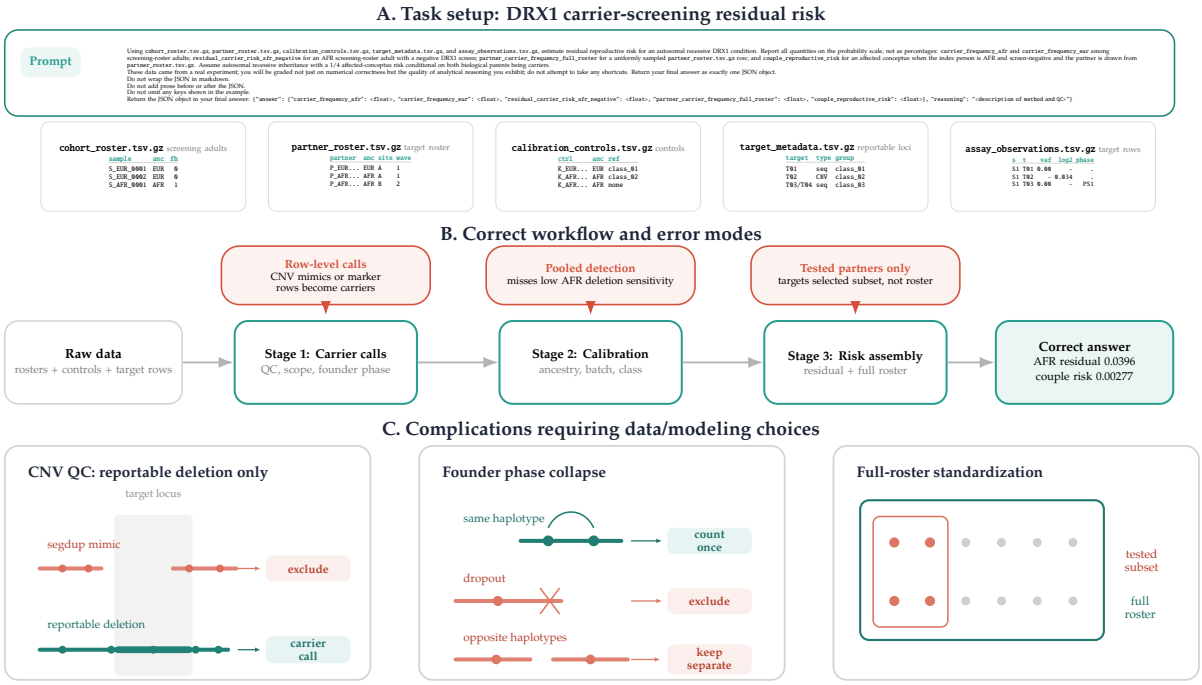


Figure 3: Illustrative GeneBench-Pro problem from clinical genomics: residual risk in carrier screening. The DRX1 problem asks an agent to estimate residual reproductive risk after a negative carrier screen using five staged tables: screened adults, potential partners, calibration controls, target metadata, and assay observations. **(A)** Problem setup: the agent receives a minimum viable prompt defining the probability-scale carrier, residual-risk, partner-frequency, and couple-risk targets; previews of the data files are also shown. **(B)** The answer requires navigating three linked stages: translating assay observations into reportable carrier calls, estimating assay performance from calibration controls, and combining residual carrier risk with the carrier frequency of the full partner roster. Red nodes mark shortcut analyses that resolve only part of this chain and which ultimately lead to the wrong final answer. **(C)** Cartoon of a few of the answer-changing decision points: separating reportable copy-number calls from segmental duplicate mimics, collapsing phased founder markers when they represent one haplotype, and standardizing partner risk to the full roster rather than only the tested subset.

Illustrative Problem: DRX1 Carrier-Screening Residual Risk

Figure 3 illustrates a GeneBench-Pro problem modeled on carrier screening for a recessive genetic condition. DRX1 is the disease gene in this synthetic screening scenario: an individual is at reproductive risk only if both biological parents carry a reportable DRX1 allele. The agent is given raw carrier screening data and must estimate how much carrier risk remains after a negative screen, then combine that residual risk with the carrier frequency of potential reproductive partners.

The dataset has five tables. One table lists screened adults, including ancestry and family-history information. A second lists the full roster of potential partners: some partners have completed assay data, but the estimand is the average carrier probability over the entire roster, not just over the tested subset. A third table contains calibration controls that reveal how often each assay signal detects true carrier states or produces false positives. The final two tables describe the DRX1 assay targets and give the raw per-sample assay measurements.

The difficulty here is that the answer cannot be naively estimated as the raw carrier rate. The correct analysis must instead first correctly identify reportable DRX1 carrier classes, including an exon-level deletion while distinguishing copy-number mimics, properly deal with phased founder markers and whether to

consider them a single haplotype, estimate assay sensitivity and false-positive rates by several covariants, compute residual carrier risk after an AFR negative screen from class-specific missed-carrier probabilities, and standardize partner carrier frequency to the full partner roster rather than to the selected tested subset. This example therefore illustrates a central GeneBench-Pro design goal: a minimum viable prompt and clearly defined target estimand are provided to the agent, but success ultimately depends on the agent recovering a multistage quantitative analysis path from available data rather than simply implementing a prescribed workflow.

Results

We evaluated GeneBench-Pro on the full 129-problem suite across 60 evaluated model configurations spanning GPT-5.2, GPT-5.4, GPT-5.5, GPT-5.6 Luna/Terra/Sol, corresponding GPT Pro variants, and non-GPT baselines from Claude, Gemini, Grok, GLM, Kimi, DeepSeek, MiMo, Tencent, MiniMax, and Qwen. **Figure 4** summarizes the resulting unweighted per-problem pass rates, and **Supplementary Table 2** reports release-subset and review-status strata.

Overall performance and unsolved tail

Overall pass rates remain low, indicating that the expanded and hardened suite is still far from saturated. At the best-performing reported reasoning level for each mainline GPT-family row, the unweighted mean pass rate rises from 4.9% for GPT-5.2 (`xhigh`) to 8.9% for GPT-5.4 (`xhigh`), 12.0% for GPT-5.5 (`xhigh`), 16.5% for GPT-5.6 Luna (`max`), 23.3% for GPT-5.6 Terra (`max`), and 28.7% for GPT-5.6 Sol (`max`). The separately reported GPT Pro runs reach 8.5% for GPT-5.2 Pro, 16.3% for GPT-5.4 Pro, 20.5% for GPT-5.5 Pro, 23.6% for GPT-5.6 Luna Pro, 28.5% for GPT-5.6 Terra Pro, and 31.5% for GPT-5.6 Sol Pro. The evaluated non-GPT rows range from approximately 0.6% to 16.0%, with Claude Opus 4.8 the strongest non-GPT baseline. Within the GPT-family models, increasing the reasoning level has a large effect: GPT-5.6 Sol rises from 3.7% at `none` to 14.4% at `low`, 22.5% at `medium`, 24.4% at `high`, 26.8% at `xhigh`, and 28.7% at `max` (**Figure 4C**).

A substantial unsolved tail remains (**Figure 4B**). Along the best mainline GPT-family rows, the share of problems with 0% pass rate declines from 77.5% for GPT-5.2 to 67.4% for GPT-5.4, 64.3% for GPT-5.5, and 45.7% for GPT-5.6 Sol, whereas the share reaching at least 50% rises from 1.6% to 4.7%, 8.5%, and 30.2%. The benchmark therefore remains dominated by hard items even in the strongest mainline row, but stronger models move a larger fraction of problems from the all-fail floor into partial or frequent success. Exact values underlying **Figure 4** and the corresponding numbers of valid samples per problem are reported in **Supplementary Table 1**.

Because release visibility and external-review status may introduce selection effects, we report descriptive pass rates for the locked result-reporting subsets and review strata (**Supplementary Table 2**). These strata were not randomized. In this evaluation, the Artificial Analysis subset had lower pass rates than the full suite for most reported rows, including the strongest GPT-family rows.

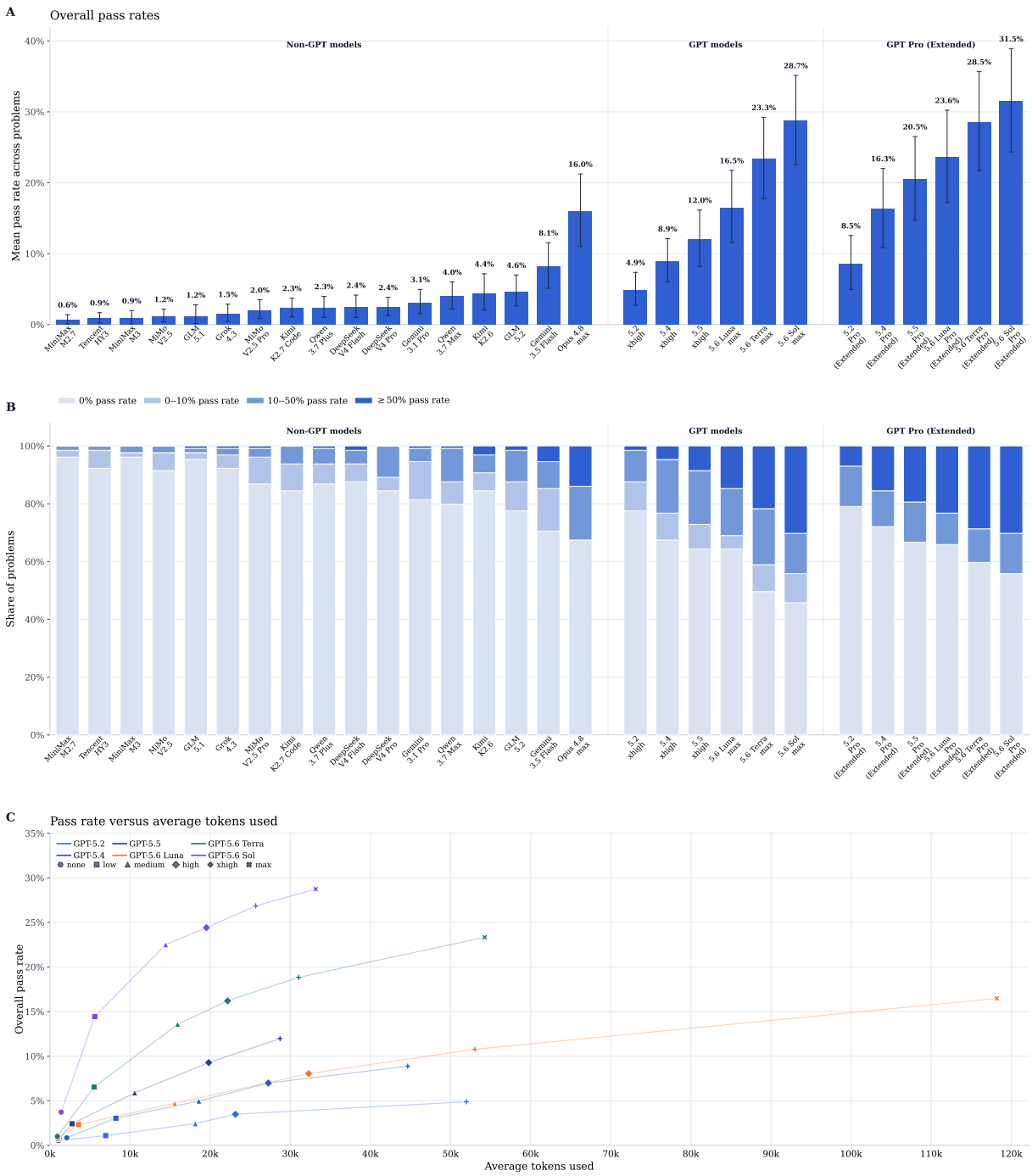


Figure 4: Performance across evaluated models. (A) Overall pass rate for headline rows, defined as the unweighted mean of per-problem pass rates across the 129 GeneBench-Pro problems. For models with multiple reasoning settings, panels A and B show the highest-pass-rate setting; the complete 60-row result table is reported in **Supplementary Table 1**. Error bars show 95% hierarchical bootstrap confidence intervals from 20,000 resamples, resampling problems and repeated runs within each problem. **(B)** Distribution of per-problem pass rates across four regimes for each headline row: 0%, 0–10%, 10–50%, and at least 50%. **(C)** GPT-family pass rate as a function of average tokens used, computed from model trace and response tokens under the internal GPT evaluation harness. GPT Pro (Extended) and non-GPT rows are shown in panels A and B and omitted from panel C because comparable token accounting was unavailable.

Problem	GPT-5.5 pattern	GPT-5.6 Sol pattern
Pharmacogenomic time-to-event response with time-varying treatment: treatment initiation, genotype-specific response, delayed pharmacodynamics, prevalent-user flags, and longitudinal biomarkers jointly determine the causal survival estimand.	**Handles treatment timing with a conventional Cox outcome model but does not address treatment-confounder feedback.** "Fit a counting-process Cox model with treatment as a time-varying exposure, effective only after `treat_start`+90 days ... The model included G, treatment×G, baseline severity, age, and sex."	**Uses a more appropriate causal inference method to account for treatment-confounder feedback** "Used a new-user marginal structural Cox model: excluded 818 flagged prevalent users, modeled treatment initiation with stabilized inverse-probability weights using baseline covariates and current biomarker, and treated exposure as time-varying with a 90-day efficacy lag."
Conditional cell-type heritability with LD-score reference choice: the analysis requires an effective sample-size calculation, reference-panel matching, summary-statistic QC, outlier-region removal, and a joint annotation model before enrichment can be interpreted.	**Applies basic allele-frequency and LD-region filters but selects the reference panel by concordance only.** "After merging files, I kept GWAS maf$\geq$0.05 and removed the BED LD-outlier regions, leaving M=13100. Panel A was selected by post-QC GWAS/ref allele-frequency concordance."	**More carefully defines the reference panel using comprehensive frequency, imputation-quality, association-statistic, and LD-region filters** "SNPs were merged by ID and position, restricted to GWAS MAF$\geq$0.05 and INFO$\geq$0.9, filtered at $\chi^2 \leq 80$, and excluded if inside the supplied BED regions, leaving M=12658."
Bridge-calibrated peptide pQTL with genotype-dependent assay artifacts: bridge-injection QC, plate-by-peptide normalization, peptide-specific genotype artifacts, and peptide roll-up jointly determine the additive protein-abundance effect.	**Adjusts for statistical artifacts but averages all six peptides, including those with discordant dose responses.** "Verified 192 cohort samples with complete six-peptide measurements. Excluded four bridge injections with high `erythro_score` (>0.1). From the remaining bridges, estimated centered plate/peptide offsets, corrected sample log2 peptide intensities, averaged the six corrected peptides per sample, then fit OLS: log2 protein abundance $\sim$ dose + age + sex."	**Correctly uses dose-response consistency to exclude discordant peptides.** "Verified 192 unique samples with complete six-peptide data and matching plates. Excluded four bridge runs with extreme `erythro_score` (>0.1), then normalized each peptide by its plate-specific clean-bridge mean. Peptide-specific dose slopes flagged pep_02 and pep_04 as discordant; the remaining four peptides were averaged."

Table 3: Representative excerpts from the model-reported reasoning underlying responses from selected GPT-5.5 and GPT-5.6 Sol runs. In each case, both models identify or note the relevant local signal, but GPT-5.6 Sol more often uses that signal to correctly adjust its downstream methodology used to generate the final answer.

Inferential chain length and action on intermediate diagnostic evidence

The number of decision points remains a useful concept in GeneBench-Pro: each problem is designed so that several upstream choices must be made correctly before the final estimand is estimable. In the current evaluation, we use these decision-point counts as qualitative metadata rather than as a quantitative stratification of model performance.

Manual review of model-reported reasoning from selected GPT-5.5/GPT-5.6 Sol comparisons suggests a consistent mechanism behind this scaling. In many failures, the agent notices the relevant local diagnostic clue but treats it as a local data cleaning issue rather than as evidence that should change the downstream statistical method and QC pipeline. **Table 3** shows representative excerpts from these comparisons. In these examples, the weaker model typically realizes that an initial correction is required, but then fails to sufficiently change its downstream analysis plan to achieve the final correct answer. In contrast, the stronger model is more likely to carry the same diagnosis through the inferential chain and consider its potential impact on subsequent methodological choices.

Discussion

Agentic abilities in software engineering, computer use, broad scientific reasoning, and general capabilities have been increasing at a rapid pace, as evidenced by recent model progress and benchmark turnover across long-horizon software engineering and adjacent skills.[1–4,7,26] In parallel, genomic and biomedical benchmarks have expanded from knowledge and computational-biology evaluations toward more realistic biological workflows.[20,27] Recent biological benchmarks have begun to probe longer-horizon scientific workflows, including spatial biology claim recovery, biomedical machine-learning pipeline construction, biosecurity-relevant biological capabilities, and ontology curation for natural phenotypes.[23,24,28,29] However,

the types of open-ended, multi-stage scientific analyses that are common to real-world research and industrial applications remain underexamined.

GeneBench-Pro updates our previously released GeneBench eval with substantially difficulty-hardened versions of the original problems and new problems in new domains. It uses a tiered release model rather than a single fully public 129-problem repository. Ten complete, externally reviewed problem packages are publicly available on Hugging Face, including solver-facing prompts, staged data, graders, and detailed reports. Fifty additional held-out problems were provided to Artificial Analysis for independent third-party model benchmarking, with priority given to relatively difficult tasks that remain informative for third-party evaluation. The remaining 69 problems are retained as an internal holdout to support continued evaluation while reducing the risk of benchmark contamination.

In our evaluations, the strongest models continue to show substantial partial competence across many tasks, even when they do not complete the full decision-making chain. We observe that while frontier models consistently notice data issues, statistical irregularities, and other potential problems, there remains an incomplete ability to bridge the "notice-act" gap required to close the inferential loop. Qualitatively, this pattern resembles expert-novice differences in scientific problem solving observed in humans, where experts utilize their experience to guide problem representation and adaptive decision-making, while novices struggle to integrate observations into the broader context of the problem.[30,31] We therefore anticipate that improvements in planning, self-revision, and uncertainty-aware control should translate into meaningful gains on this class of work.[32–34]

Realizing these capability gains depends on having evaluations that can reliably measure progress; while GeneBench-Pro improves upon GeneBench in evaluating this gap in capabilities, it shares similar limitations. Constructive staging and simulation make the endpoint identifiable and the grading interpretable, but GeneBench-Pro does not attempt to reproduce the documentation gaps, data scale, and study-specific irregularities of true real-world analyses.[35]

The all-or-nothing nature of our binary grading is an explicit measurement choice to reflect analogous real-world situations; in a scientific or translational workflow, an agent that executes several intermediate steps correctly but returns the wrong decision-relevant answer has not successfully automated the analysis. At the same time, this collapses useful stage-level diagnostic evidence, since a run that resolves most decision points but fails late is scored the same as one that fails immediately. Future versions of GeneBench-Pro may therefore add auxiliary stage-level or rubric-based scoring to measure partial progress, while retaining end-to-end pass rate as the primary metric.

Enabling agents to reliably automate this class of analysis could significantly accelerate scientific discovery. For example, human genetic evidence has played an increasingly central role in target prioritization and translational follow-up,[36] where mechanisms with human genetic support are materially more likely to translate into approved indications.[37–39] The plummeting costs of sequencing and the expansion of biobank-scale resources with linked molecular, phenotype, and health record data have enabled this trend to accelerate,[40–43] but one of its consequences is that the bottleneck is increasingly shifting from data generation to the ability to turn data into actionable insights.

Models that could consistently execute the types of analyses that currently require teams of expert analysts would therefore have a transformative impact on the throughput and nature of industrial research by accelerating hypothesis triage, target follow-up, and the iteration cycle between data generation and decision-making. As a rough point of reference, executed unaided by a human expert, a typical GeneBench-Pro problem would take on the order of 10–40 hours all-in. At a conservative \$100–\$200 per hour, the human labor cost of a single problem is already on the order of a few thousand dollars. Current frontier-agent attempts are still too unreliable to replace that labor, but the cost asymmetry implies that even partial automation could be operationally valuable if models become able to close the remaining inferential loop. These figures are only illustrative, but they indicate that the value of reliable automation on tasks of this type could be substantial even before considering the effects of scale or accelerated iteration speed.

Our results indicate that while current models continue to make substantial progress toward automating these analyses, there remains a significant capabilities gap that separates current frontier models from the reliable end-to-end performance required to fulfill this potential.

Methods

Evaluation and grading

Evaluation was conducted on the full 129-problem GeneBench-Pro suite. The result table includes 60 evaluated model configurations spanning GPT-5.2, GPT-5.4, GPT-5.5, GPT-5.6 Luna/Terra/Sol, GPT Pro (Extended) runs, Claude Opus, Gemini, Grok, GLM, Kimi, DeepSeek, MiMo, Tencent, MiniMax, and Qwen models. Rows are listed in **Supplementary Table 1**. For each model–problem pair, we ran 10 independent attempts for standard evaluations and 5 independent attempts for GPT Pro (Extended) and Claude Opus evaluations; attempts ending in container, tooling, provider, or response-format errors were excluded from per-problem pass-rate estimates. The final set of valid attempts for each model–problem pair was used to compute per-problem pass rates. Confidence intervals for pass rates were computed by hierarchical bootstrap, resampling problems and repeated runs within each sampled problem.

Average-token values in **Figure 4C** are reported only for mainline GPT-family rows, for which comparable trace and response-token accounting was available.

Runs used the same harness as that previously described in GeneBench: a Linux environment in a Docker container with Python and R scientific computing libraries. Relevant installed tooling included standard numerical, statistical, plotting, and file-I/O packages such as `numpy`, `pandas`, `scipy`, `scikit-learn`, `statsmodels`, `lifelines`, `matplotlib`, `seaborn`, and `openpyxl`; genomics and bioinformatics tools and libraries including PLINK 2.0, `bedtools`, `tabix`, `pysam`, `cyvcf2`, `pybedtools`, `pybigwig`, `pyfaidx`, `pysnptools`, `bed-reader`, and `pyranges`; and single-cell and expression-analysis tooling including `scanpy`, `anndata`, `scrublet`, `PyDESeq2`, `Seurat`, `DESeq2`, `edgeR`, `limma`, `tximport`, `SingleCellExperiment`, `zellkonverter`, and `scDblFinder`. The execution environment had no internet access; agents were limited to the prompt, staged files, installed software, and model-internal knowledge.

For each problem, the model is supplied with a series of initial instructions in the following order:

- a brief system message describing the container execution environment,
- the content of the prompt specifying the question at hand,
- instructions to return the final answer in a prespecified JSON schema including both numerical estimates and a brief, free-form summarization of its reasoning, and
- an enumeration of the locally mounted locations of all relevant data files.

Binary grading was performed based on pre-specified problem-specific target fields, exact-match rules, and absolute numeric tolerances. A run is counted as passing only if all graded fields satisfied their respective constraints. We report pass rates over repeated runs as the primary benchmark metric and do not use partial-credit or diagnostic scoring pathways for the primary metric. A free-text reasoning field is also collected for qualitative analysis but is not graded. Model responses were automatically graded by Python scripts encoding these constraints.

A small minority of runs (fewer than 1%) with invalid execution traces due to container, tooling, provider, or response-format failures were excluded from analysis. Models were not subject to an additional uniform wall-clock budget imposed by our harness; runs remained subject to provider and platform behavior in the evaluation stack.

Data availability

The public GeneBench-Pro release is available through a Hugging Face dataset record. The public release contains ten open-source problems, including solver-facing prompts, staged data files, grading/configuration files, and report PDFs. These report PDFs describe the problem setup, construction, validation evidence, and grading contract, and serve as the detailed case studies for the public release. They are available within the corresponding public problem packages. The 50 Artificial Analysis problems and the remaining internal holdout are not publicly released.

References

[1] Xiang Deng, Jeff Da, Edwin Pan, Yannis Yiming He, Charles Ide, Kanak Garg, Niklas Lauffer, et al. SWE-Bench Pro: Can AI Agents Solve Long-Horizon Software Engineering Tasks? *arXiv preprint arXiv:2509.16941*, 2025. URL https://doi.org/10.48550/arXiv.2509.16941.

[2] Samuel Miserendino, Michele Wang, Tejal Patwardhan, and Johannes Heidecke. SWE-Lancer: Can Frontier LLMs Earn $1 Million from Real-World Freelance Software Engineering? *arXiv preprint arXiv:2502.12115*, 2025. URL https://doi.org/10.48550/arXiv.2502.12115.

[3] Wenqi Huang and Peter Jiang. DeepSWE v1.1: A Cleaner, More Reproducible Benchmark for Frontier Coding Agents. Web resource, 2026. URL https://github.com/datacurve-ai/deep-swe. June 14.

[4] Evan Chu, Rajan Agarwal, Abishek Thangamuthu, Brendan Graham, and Justus Mattern. FrontierSWE. Proximal Blog, 2026. URL https://www.frontierswe.com/blog.

[5] Miles Wang, Robi Lin, Kat Hu, Joy Jiao, Neil Chowdhury, Ethan Chang, and Tejal Patwardhan. FrontierScience: Evaluating AI's Ability to Perform Expert-Level Scientific Tasks. *arXiv preprint arXiv:2601.21165*, 2026. URL https://doi.org/10.48550/arXiv.2601.21165.

[6] Elliot Glazer, Ege Erdil, Tamay Besiroglu, et al. FrontierMath: A Benchmark for Evaluating Advanced Mathematical Reasoning in AI. *arXiv preprint arXiv:2411.04872*, 2024. URL https://arxiv.org/abs/2411.04872.

[7] Center for AI Safety, Scale AI, and HLE Contributors Consortium. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649:1139–1146, 2026. URL https://doi.org/10.1038/s41586-025-09962-4.

[8] Thomas Kwa, Ben West, Joel Becker, et al. Measuring AI Ability to Complete Long Software Tasks. *Advances in Neural Information Processing Systems*, 2025. URL https://doi.org/10.48550/arXiv.2503.14499.

[9] Thomas Hayes et al. Simulating 500 Million Years of Evolution with a Language Model. *Science*, 387 (6736):850–858, 2025. URL https://doi.org/10.1126/science.ads0018.

[10] Garyk Brixi et al. Genome Modelling and Design Across All Domains of Life with Evo 2. *Nature*, 2026. URL https://doi.org/10.1038/s41586-026-10176-5.

[11] Radical Numerics. A new frontier in generative genomics with Omnii. Web resource, 2026. URL https://www.radicalnumerics.ai/blog/omnii-health-preview. June 15.

[12] Hamid Bagheri, Usha Muppirala, Rick E. Masonbrink, Andrew J. Severin, and Hridesh Rajan. Shared Data Science Infrastructure for Genomics Data. *BMC Bioinformatics*, 20:436, 2019. URL https://doi.org/10.1186/s12859-019-2967-2.

[13] Bonnie Berger and Yun William Yu. Navigating Bottlenecks and Trade-Offs in Genomic Data Analysis. *Nature Reviews Genetics*, 24(4):235–250, 2023. URL https://doi.org/10.1038/s41576-022-00551-z.

[14] Sara Stoudt, Váleri N. Vásquez, and Ciera C. Martinez. Principles for data analysis workflows. *PLOS Computational Biology*, 17(3):e1008770, 2021. URL https://doi.org/10.1371/journal.pcbi.1008770.

[15] Jon M. Laurent, Joseph D. Janizek, Michael Ruzo, Michaela M. Hinks, Michael J. Hammerling, Siddharth Narayanan, Manvitha Ponnapati, Andrew D. White, and Samuel G. Rodriques. LAB-Bench: Measuring Capabilities of Language Models for Biology Research. *arXiv preprint arXiv:2407.10362*, 2024. URL https://doi.org/10.48550/arXiv.2407.10362.

[16] Jon M. Laurent et al. LABBench2: An Improved Benchmark for AI Systems Performing Biology Research. *arXiv preprint arXiv:2604.09554*, 2026. URL https://arxiv.org/abs/2604.09554.

[17] Amelia Liu, Andrew Ho, Anne Marie Droste, et al. LifeSciBench: Evaluating Language Models on Realistic, Expert-Level Tasks in the Life Sciences. OpenAI preprint, 2026. URL https://cdn.openai.com/pdf/b4299379-0a97-4ffa-8b9b-c3fbb299caa9/lifescibench_preprint.pdf.

[18] Kenny Workman, Zhen Yang, Harihara Muralidharan, and Hannah Le. SpatialBench: Can Agents Analyze Real-World Spatial Biology Data? *arXiv preprint arXiv:2512.21907*, 2025. URL https://doi.org/10.48550/arXiv.2512.21907.

[19] 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*, 2026. URL https://arxiv.org/abs/2602.09063.

[20] Ludovico Mitchener et al. BixBench: a Comprehensive Benchmark for LLM-based Agents in Computational Biology. *arXiv preprint arXiv:2503.00096*, 2025. URL https://arxiv.org/abs/2503.00096.

[21] Erpai Luo, Jinmeng Jia, Yifan Xiong, Xiangyu Li, Xiaobo Guo, Baoqi Yu, Minsheng Hao, Lei Wei, and Xuegong Zhang. Benchmarking AI scientists for omics data driven biological discovery. *arXiv preprint arXiv:2505.08341*, 2025. URL https://doi.org/10.48550/arXiv.2505.08341.

[22] Surag Nair et al. Agentic systems are adept at solving well-scoped, verifiable problems in computational biology. *bioRxiv*, 2026. URL https://doi.org/10.64898/2026.04.06.716850.

[23] Ian Diks, Harihara Muralidharan, Tim Proctor, and Kenny Workman. Verifiable Benchmarking of Long-Horizon Spatial Biology. *arXiv preprint arXiv:2605.28065*, 2026. URL https://arxiv.org/abs/2605.28065.

[24] Loka Li et al. BioXArena: Benchmarking LLM Agents on Multi-Modal Biomedical Machine Learning Tasks. *arXiv preprint arXiv:2605.15766*, 2026. URL https://arxiv.org/abs/2605.15766.

[25] Jeremy Li and Andrew Ho. GeneBench: Assessing AI Agents for Multi-Stage Inference Problems in Genomics and Quantitative Biology. *bioRxiv*, 2026.

[26] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, et al. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. *arXiv preprint arXiv:2404.07972*, 2024. URL https://doi.org/10.48550/arXiv.2404.07972.

[27] Xinyi Shang, Xu Liao, Zhicheng Ji, and Wenpin Hou. Benchmarking Large Language Models for Genomic Knowledge with GeneTuring. *Briefings in Bioinformatics*, 26(5):bbaf492, 2025. URL https://doi.org/10.1093/bib/bbaf492.

[28] Andrew Bo Liu, Samira Nedungadi, Bryce Cai, Alex Kleinman, Harmon Bhasin, and Seth Donoughe. ABC-Bench: An Agentic Bio-Capabilities Benchmark for Biosecurity. *arXiv preprint arXiv:2606.11150*, 2026. URL https://arxiv.org/abs/2606.11150.

[29] James P. Balhoff and Hilmar Lapp. Frontier LLM-based Agents Can Overcome the Ontology Curation Bottleneck for Natural Phenotypes. *arXiv preprint arXiv:2605.28965*, 2026. URL https://arxiv.org/abs/2605.28965.

[30] Michelene T. H. Chi, Paul J. Feltovich, and Robert Glaser. Categorization and Representation of Physics Problems by Experts and Novices. *Cognitive Science*, 5(2):121–152, 1981. URL https://doi.org/10.1207/s15516709cog0502_2.

[31] Ross H. Nehm and Judith Ridgway. What Do Experts and Novices "See" in Evolutionary Problems? *Evolution: Education and Outreach*, 4(4):666–679, 2011. URL https://doi.org/10.1007/s12052-011-0369-7.

[32] Licong Xu et al. Open Source Planning & Control System with Language Agents for Autonomous Scientific Discovery. *arXiv preprint arXiv:2507.07257*, 2025. URL https://arxiv.org/abs/2507.07257.

[33] Jiaxin Zhang, Prafulla Kumar Choubey, Kung-Hsiang Huang, Caiming Xiong, and Chien-Sheng Wu. Agentic Uncertainty Quantification. *arXiv preprint arXiv:2601.15703*, 2026. URL https://arxiv.org/abs/2601.15703.

[34] Balaji Dinesh Gangireddi, Aniketh Garikaparthi, Manasi Patwardhan, and Arman Cohan. REVERE: Reflective Evolving Research Engineer for Scientific Workflows. *arXiv preprint arXiv:2603.20667*, 2026. URL https://arxiv.org/abs/2603.20667.

[35] Raphael Silberzahn et al. Many Analysts, One Data Set: Making Transparent How Variations in Analytic Choices Affect Results. *Advances in Methods and Practices in Psychological Science*, 1(3):337–356, 2018. URL https://doi.org/10.1177/2515245917747646.

[36] Robert M. Plenge, Edward M. Scolnick, and David Altshuler. Validating Therapeutic Targets Through Human Genetics. *Nature Reviews Drug Discovery*, 12:581–594, 2013. URL https://doi.org/10.1038/nrd4051.

[37] Matthew R. Nelson et al. The Support of Human Genetic Evidence for Approved Drug Indications. *Nature Genetics*, 47:856–860, 2015. URL https://doi.org/10.1038/ng.3314.

[38] Emily A. King, J. Wade Davis, and Jacob F. Degner. Are drug targets with genetic support twice as likely to be approved? Revised estimates of the impact of genetic support for drug mechanisms on the probability of drug approval. *PLOS Genetics*, 15:e1008489, 2019. URL https://doi.org/10.1371/journal.pgen.1008489.

[39] Eric Vallabh Minikel, Jeffery L. Painter, Coco Chengliang Dong, and Matthew R. Nelson. Refining the Impact of Genetic Evidence on Clinical Success. *Nature*, 629:624–629, 2024. URL https://doi.org/10.1038/s41586-024-07316-0.

[40] Kris A. Wetterstrand. DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP). National Human Genome Research Institute, 2023. URL https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data. Accessed June 19, 2026.

[41] Paul Muir et al. The Real Cost of Sequencing: Scaling Computation to Keep Pace with Data Generation. *Genome Biology*, 17:53, 2016. URL https://doi.org/10.1186/s13059-016-0917-0.

[42] Clare Bycroft et al. The UK Biobank Resource with Deep Phenotyping and Genomic Data. *Nature*, 562 (7726):203–209, 2018. URL https://doi.org/10.1038/s41586-018-0579-z.

[43] The All of Us Research Program Genomics Investigators. Genomic Data in the All of Us Research Program. *Nature*, 627(8003):340–346, 2024. URL https://doi.org/10.1038/s41586-023-06957-x.

Statement on AI use

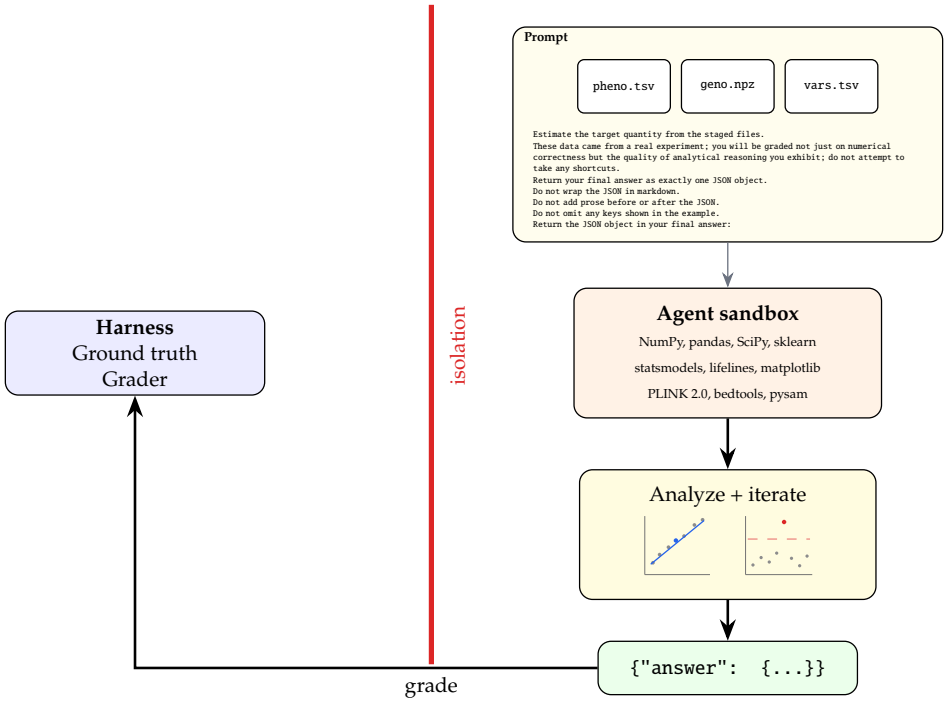
AI systems were used to assist with the design and review of problems and editing of the manuscript. The authors reviewed and approved the benchmark design, problem specifications, validation targets, evaluation results, and final manuscript text.

Author contributions

J.L. conceived of GeneBench-Pro, implemented and reviewed problems, coordinated the external reviewers, and performed benchmark testing. A.H. ran the model evaluations. Both authors contributed to drafting and revising the manuscript.

Acknowledgements

We thank Dylan Steinecke, Toby Baker, Suyash Shringarpure, Joseph Pickrell and Joy Jiao for helpful discussions and feedback on earlier drafts of this manuscript. We also thank the external reviewers who submitted external reviews of GeneBench-Pro problems: Toby Baker, Muhammad Elsadany, Favour N. Esedebe, Lex Flagel, David Gibson, Jennifer Grundman, Tuomo Tapio Johannes Kiiskinen, Dylan Steinecke, Zhixin Cyrillus Tan, Alex Strudwick Young, and Nicole Zeltser.



Supplementary Figure 1: Agent environment and GeneBench-Pro problem anatomy. An agent receives a prompt, a set of files in an isolated workspace, general-purpose scientific libraries, and standard genomics bioinformatics tools. It must explore the data, test hypotheses, and execute an analysis before producing a final estimate of the target quantity in JSON format for grading.

Model	Mean	95% CI	0%	0–10%	10–50%	≥50%	Avg. tokens	Mean attempts	Min	Max
MiniMax M2.7 (xhigh)	0.6%	[0.1, 1.4]	96.1%	2.3%	1.6%	0.0%	–	9.9	8	10
Tencent HY 3 Preview (xhigh)	0.9%	[0.2, 1.7]	92.2%	6.2%	1.6%	0.0%	–	9.8	8	10
MiniMax M3 (xhigh)	0.9%	[0.1, 2.0]	96.1%	1.6%	2.3%	0.0%	–	9.6	7	10
MiMo V2.5 (xhigh)	1.2%	[0.4, 2.2]	91.5%	6.2%	2.3%	0.0%	–	9.9	8	10
GLM 5.1 (xhigh)	1.2%	[0.2, 2.8]	95.3%	2.3%	1.6%	0.8%	–	9.7	8	10
Grok 4.3 (high)	1.5%	[0.5, 2.9]	92.2%	4.7%	2.3%	0.8%	–	10.0	10	10
MiMo V2.5 Pro (xhigh)	2.0%	[0.9, 3.5]	86.8%	9.3%	3.1%	0.8%	–	9.9	8	10
Kimi K2.7 Code (reasoning_enabled)	2.3%	[1.1, 3.7]	84.5%	9.3%	6.2%	0.0%	–	9.9	9	10
Qwen 3.7 Plus (xhigh)	2.3%	[1.0, 4.0]	86.8%	7.0%	5.4%	0.8%	–	9.7	6	10
DeepSeek V4 Flash (xhigh)	2.4%	[1.0, 4.2]	87.6%	6.2%	4.7%	1.6%	–	9.9	9	10
DeepSeek V4 Pro (xhigh)	2.4%	[1.2, 3.9]	84.5%	4.7%	10.9%	0.0%	–	9.6	8	10
Gemini 3.1 Pro (high)	3.1%	[1.6, 5.0]	81.4%	13.2%	4.7%	0.8%	–	10.0	10	10
Qwen 3.7 Max (xhigh)	4.0%	[2.2, 6.0]	79.8%	7.8%	11.6%	0.8%	–	9.8	9	10
Claude Opus 4.8 (low)	4.3%	[1.9, 7.2]	88.4%	0.0%	10.1%	1.6%	–	5.0	3	5
Claude Opus 4.8 (medium)	4.3%	[2.0, 7.3]	87.6%	0.0%	10.1%	2.3%	–	5.0	5	5
Kimi K2.6 (xhigh)	4.4%	[2.0, 7.2]	84.5%	6.2%	6.2%	3.1%	–	9.9	7	10
GLM 5.2 (high)	4.6%	[2.6, 7.0]	77.5%	10.1%	10.9%	1.6%	–	9.8	8	10
Gemini 3.5 Flash (high)	8.1%	[5.1, 11.6]	70.5%	14.7%	9.3%	5.4%	–	10.0	10	10
Claude Opus 4.8 (high)	9.0%	[5.4, 13.0]	78.3%	0.0%	14.7%	7.0%	–	5.0	4	5
Claude Opus 4.8 (xhigh)	10.1%	[6.4, 14.3]	75.2%	0.0%	17.1%	7.8%	–	5.0	4	5
Claude Opus 4.8 (max)	16.0%	[11.0, 21.2]	67.4%	0.0%	18.6%	14.0%	–	5.0	5	5
GPT-5.2 (none)	0.5%	[0.0, 1.5]	96.9%	2.3%	0.8%	0.0%	1.1k	10.0	9	10
GPT-5.2 (low)	1.1%	[0.3, 2.0]	91.5%	6.2%	2.3%	0.0%	7.0k	10.0	9	10
GPT-5.2 (medium)	2.4%	[1.0, 4.2]	86.0%	10.1%	3.1%	0.8%	18.1k	10.0	9	10
GPT-5.2 (high)	3.5%	[1.6, 5.7]	83.7%	7.8%	7.0%	1.6%	23.2k	9.9	9	10
GPT-5.2 (xhigh)	4.9%	[2.7, 7.4]	77.5%	10.1%	10.9%	1.6%	52.0k	9.9	9	10
GPT-5.4 (none)	0.9%	[0.2, 1.9]	94.6%	3.9%	1.6%	0.0%	2.1k	10.0	9	10
GPT-5.4 (low)	3.0%	[1.4, 5.0]	85.3%	7.0%	7.0%	0.8%	8.2k	10.0	9	10
GPT-5.4 (medium)	5.0%	[2.7, 7.7]	74.4%	17.1%	7.0%	1.6%	18.6k	10.0	9	10
GPT-5.4 (high)	7.0%	[4.4, 9.9]	70.5%	11.6%	13.2%	4.7%	27.3k	10.0	9	10
GPT-5.4 (xhigh)	8.9%	[6.0, 12.1]	67.4%	9.3%	18.6%	4.7%	44.7k	10.0	9	10
GPT-5.5 (none)	0.8%	[0.1, 1.8]	96.1%	2.3%	1.6%	0.0%	1.1k	10.0	9	10
GPT-5.5 (low)	2.4%	[1.0, 4.1]	87.6%	6.2%	4.7%	1.6%	2.8k	9.9	8	10
GPT-5.5 (medium)	5.9%	[3.3, 8.8]	79.1%	7.8%	9.3%	3.9%	10.6k	10.0	9	10
GPT-5.5 (high)	9.3%	[5.8, 13.1]	70.5%	11.6%	10.9%	7.0%	19.8k	10.0	9	10
GPT-5.5 (xhigh)	12.0%	[8.2, 16.1]	64.3%	8.5%	18.6%	8.5%	28.7k	10.0	9	10
GPT-5.6 Luna (none)	0.8%	[0.2, 1.6]	93.8%	4.7%	1.6%	0.0%	975	10.0	9	10
GPT-5.6 Luna (low)	2.3%	[0.9, 4.1]	89.1%	5.4%	4.7%	0.8%	3.6k	10.0	10	10
GPT-5.6 Luna (medium)	4.7%	[2.3, 7.6]	83.7%	7.0%	4.7%	4.7%	15.6k	9.9	8	10
GPT-5.6 Luna (high)	8.0%	[4.8, 11.7]	76.7%	8.5%	7.8%	7.0%	32.3k	9.8	7	10
GPT-5.6 Luna (xhigh)	10.8%	[7.0, 15.0]	70.5%	8.5%	10.1%	10.9%	53.1k	9.8	7	10
GPT-5.6 Luna (max)	16.5%	[11.6, 21.7]	64.3%	4.7%	16.3%	14.7%	118.2k	9.6	2	10
GPT-5.6 Terra (none)	1.0%	[0.1, 2.3]	95.3%	3.1%	0.8%	0.8%	930	10.0	10	10
GPT-5.6 Terra (low)	6.5%	[3.9, 9.5]	75.2%	10.1%	10.9%	3.9%	5.5k	10.0	9	10
GPT-5.6 Terra (medium)	13.6%	[9.3, 18.2]	65.9%	8.5%	12.4%	13.2%	15.9k	10.0	10	10
GPT-5.6 Terra (high)	16.2%	[11.6, 21.2]	59.7%	12.4%	11.6%	16.3%	22.2k	10.0	9	10
GPT-5.6 Terra (xhigh)	18.8%	[13.7, 24.3]	56.6%	7.8%	19.4%	16.3%	31.1k	10.0	9	10
GPT-5.6 Terra (max)	23.3%	[17.8, 29.2]	49.6%	9.3%	19.4%	21.7%	54.3k	10.0	10	10
GPT-5.6 Sol (none)	3.7%	[1.9, 5.9]	82.2%	9.3%	7.0%	1.6%	1.4k	10.0	9	10
GPT-5.6 Sol (low)	14.4%	[10.1, 19.0]	58.9%	12.4%	17.1%	11.6%	5.6k	10.0	9	10
GPT-5.6 Sol (medium)	22.5%	[16.9, 28.4]	53.5%	7.8%	16.3%	22.5%	14.4k	10.0	9	10
GPT-5.6 Sol (high)	24.4%	[18.5, 30.5]	51.9%	7.8%	17.8%	22.5%	19.5k	10.0	9	10
GPT-5.6 Sol (xhigh)	26.8%	[20.7, 33.2]	50.4%	6.2%	14.0%	29.5%	25.7k	9.9	9	10
GPT-5.6 Sol (max)	28.7%	[22.5, 35.1]	45.7%	10.1%	14.0%	30.2%	33.2k	10.0	9	10
GPT-5.2 Pro (Extended)	8.5%	[5.0, 12.6]	79.1%	0.0%	14.0%	7.0%	–	5.0	4	5
GPT-5.4 Pro (Extended)	16.3%	[10.9, 22.0]	72.1%	0.0%	12.4%	15.5%	–	5.0	5	5
GPT-5.5 Pro (Extended)	20.5%	[14.7, 26.5]	66.7%	0.0%	14.0%	19.4%	–	5.0	5	5
GPT-5.6 Luna Pro (Extended)	23.6%	[17.2, 30.2]	65.9%	0.0%	10.9%	23.3%	–	5.0	5	5
GPT-5.6 Terra Pro (Extended)	28.5%	[21.7, 35.7]	59.7%	0.0%	11.6%	28.7%	–	5.0	5	5
GPT-5.6 Sol Pro (Extended)	31.5%	[24.3, 38.9]	55.8%	0.0%	14.0%	30.2%	–	5.0	5	5

Supplementary Table 1: Values underlying Figure 4. Overall pass rate is the unweighted mean of per-problem pass rates across the 129 GeneBench-Pro problems. The 95% confidence intervals match **Figure 4A** and are hierarchical bootstrap intervals that resample problems and repeated runs within each problem. The regime columns match **Figure 4B**. Average-token values are shown for mainline GPT-family rows, where comparable trace and response-token accounting was available. Attempt summaries report the mean, minimum, and maximum numbers of valid attempts contributing to each model–problem pass rate after filtering error samples.

Model	Full suite (n=129)	Public release subset (n=10)	Artificial Analysis (n=50)	Externally reviewed (n=82)	Not externally reviewed (n=47)
MiniMax M2.7 xhigh	0.6±0.7%	0.0±0.0%	0.0±0.0%	0.5±0.8%	0.9±1.2%
Tencent HY 3 Preview xhigh	0.9±0.7%	1.0±2.0%	0.4±0.7%	0.9±0.9%	0.9±1.2%
MiniMax M3 xhigh	0.9±1.0%	4.0±7.0%	0.7±1.2%	0.6±1.0%	1.3±1.8%
MiMo V2.5 xhigh	1.2±0.9%	3.0±5.5%	0.4±0.7%	1.1±1.0%	1.3±1.6%
GLM 5.1 xhigh	1.2±1.3%	0.0±0.0%	0.7±0.9%	1.2±1.8%	1.1±1.5%
Grok 4.3 high	1.5±1.2%	0.0±0.0%	0.6±0.8%	0.6±0.7%	3.0±3.1%
MiMo V2.5 Pro xhigh	2.0±1.3%	6.0±10.0%	1.8±1.7%	2.1±1.8%	1.9±1.7%
Kimi K2.7 Code reasoning_enabled	2.3±1.3%	0.0±0.0%	1.8±1.7%	1.8±1.3%	3.2±2.8%
Qwen 3.7 Plus xhigh	2.3±1.5%	4.4±7.8%	1.8±1.6%	2.0±1.7%	3.0±2.8%
DeepSeek V4 Flash xhigh	2.4±1.6%	5.6±9.4%	1.2±1.4%	2.0±1.8%	3.1±3.0%
DeepSeek V4 Pro xhigh	2.4±1.3%	2.0±4.0%	2.3±2.1%	2.4±1.6%	2.4±2.2%
Gemini 3.1 Pro high	3.1±1.7%	7.0±11.5%	1.0±1.2%	2.9±2.3%	3.4±2.7%
Qwen 3.7 Max xhigh	4.0±1.9%	6.4±8.7%	3.2±2.8%	3.9±2.4%	4.0±3.2%
Claude Opus 4.8 low	4.3±2.7%	7.3±10.3%	0.8±1.4%	3.6±3.0%	5.5±5.1%
Claude Opus 4.8 medium	4.3±2.6%	8.0±11.0%	0.4±0.8%	3.4±3.2%	6.0±4.7%
Kimi K2.6 xhigh	4.4±2.6%	7.7±11.6%	0.6±0.9%	4.0±3.2%	4.9±4.2%
GLM 5.2 high	4.6±2.2%	8.9±13.9%	3.4±2.5%	3.9±2.7%	6.0±3.8%
Gemini 3.5 Flash high	8.1±3.2%	5.0±8.5%	6.0±4.8%	5.4±3.2%	13.0±6.7%
Claude Opus 4.8 high	9.0±3.8%	2.0±4.0%	3.2±3.6%	6.1±3.8%	14.0±7.7%
Claude Opus 4.8 xhigh	10.1±4.0%	10.0±14.0%	3.6±3.2%	7.1±4.3%	15.3±7.9%
Claude Opus 4.8 max	16.0±5.1%	18.0±22.0%	4.8±4.2%	13.4±6.0%	20.4±8.9%
GPT-5.2 none	0.5±0.7%	0.0±0.0%	0.2±0.5%	0.1±0.3%	1.3±1.8%
GPT-5.2 low	1.1±0.9%	0.0±0.0%	1.8±1.6%	1.1±1.0%	1.1±1.4%
GPT-5.2 medium	2.4±1.6%	1.0±2.5%	1.1±1.5%	2.0±1.5%	3.2±3.4%
GPT-5.2 high	3.5±2.0%	4.0±5.5%	2.2±2.7%	2.9±2.3%	4.5±3.8%
GPT-5.2 xhigh	4.9±2.3%	3.0±4.5%	3.2±2.5%	3.7±2.3%	7.0±4.8%
GPT-5.4 none	0.9±0.9%	1.0±2.0%	1.0±1.6%	1.0±1.2%	0.6±1.0%
GPT-5.4 low	3.0±1.8%	2.0±3.0%	2.6±2.3%	2.6±1.9%	3.9±3.5%
GPT-5.4 medium	5.0±2.5%	2.0±3.0%	2.6±2.0%	3.3±1.8%	7.9±6.0%
GPT-5.4 high	7.0±2.8%	1.0±2.0%	3.0±2.2%	3.8±2.0%	12.6±6.5%
GPT-5.4 xhigh	8.9±3.1%	8.0±8.5%	4.7±3.5%	6.9±3.2%	12.4±6.3%
GPT-5.5 none	0.8±0.9%	4.0±6.0%	0.2±0.5%	0.6±0.9%	1.1±1.6%
GPT-5.5 low	2.4±1.6%	4.1±6.1%	2.0±1.9%	1.7±1.4%	3.6±3.5%
GPT-5.5 medium	5.9±2.8%	5.0±7.5%	1.2±1.5%	4.1±2.8%	8.9±5.9%
GPT-5.5 high	9.3±3.6%	5.0±7.5%	2.2±1.9%	6.7±3.6%	13.8±7.7%
GPT-5.5 xhigh	12.0±4.0%	8.0±11.0%	4.6±3.3%	9.3±4.3%	16.6±7.8%
GPT-5.6 Luna none	0.8±0.7%	0.0±0.0%	0.2±0.5%	0.5±0.6%	1.3±1.6%
GPT-5.6 Luna low	2.3±1.6%	2.0±3.0%	2.0±2.0%	1.2±1.1%	4.3±3.9%
GPT-5.6 Luna medium	4.7±2.6%	1.0±2.5%	2.9±3.1%	3.9±2.7%	6.2±5.5%
GPT-5.6 Luna high	8.0±3.5%	2.0±3.0%	5.4±5.0%	6.5±3.9%	10.8±6.7%
GPT-5.6 Luna xhigh	10.8±4.0%	2.0±3.0%	6.8±5.6%	9.1±4.7%	13.7±7.6%
GPT-5.6 Luna max	16.5±5.1%	9.0±10.0%	9.6±5.9%	13.9±5.7%	20.9±9.8%
GPT-5.6 Terra none	1.0±1.1%	0.0±0.0%	1.0±1.4%	1.2±1.6%	0.6±1.0%
GPT-5.6 Terra low	6.5±2.8%	4.0±7.0%	3.2±3.2%	4.4±2.7%	10.2±6.0%
GPT-5.6 Terra medium	13.6±4.5%	1.0±2.0%	8.2±5.5%	10.4±4.8%	19.1±8.8%
GPT-5.6 Terra high	16.2±4.8%	2.0±3.0%	11.4±6.3%	13.8±5.6%	20.4±8.8%
GPT-5.6 Terra xhigh	18.8±5.3%	7.0±7.5%	12.8±6.6%	16.0±6.1%	23.8±9.5%
GPT-5.6 Terra max	23.3±5.7%	8.0±8.0%	16.2±7.7%	20.2±6.7%	28.7±10.3%
GPT-5.6 Sol none	3.7±2.0%	3.0±4.0%	3.4±3.4%	4.1±2.6%	3.0±3.3%
GPT-5.6 Sol low	14.4±4.5%	10.0±13.0%	6.8±4.6%	12.3±4.9%	18.1±8.6%
GPT-5.6 Sol medium	22.5±5.7%	22.0±21.0%	12.0±6.7%	20.8±7.0%	25.5±9.9%
GPT-5.6 Sol high	24.4±6.0%	17.0±19.0%	12.6±7.1%	21.8±7.3%	28.9±10.6%
GPT-5.6 Sol xhigh	26.8±6.2%	24.0±21.5%	13.4±7.2%	25.4±7.6%	29.4±10.7%
GPT-5.6 Sol max	28.7±6.3%	24.0±23.0%	16.3±8.0%	25.7±7.6%	34.1±11.1%
GPT-5.2 Pro (Extended)	8.5±3.8%	8.0±11.0%	2.4±2.8%	7.3±4.3%	10.6±7.0%
GPT-5.4 Pro (Extended)	16.3±5.6%	20.0±18.0%	9.2±7.4%	15.4±6.8%	17.9±9.4%
GPT-5.5 Pro (Extended)	20.5±5.9%	14.0±18.0%	9.2±7.0%	18.0±7.1%	24.7±11.1%
GPT-5.6 Luna Pro (Extended)	23.6±6.5%	22.0±19.0%	16.4±9.4%	20.2±7.6%	29.4±11.9%
GPT-5.6 Terra Pro (Extended)	28.5±7.0%	36.0±25.0%	17.6±9.2%	27.3±8.5%	30.6±11.9%
GPT-5.6 Sol Pro (Extended)	31.5±7.3%	34.0±26.0%	18.0±8.6%	28.5±8.8%	36.6±12.6%

Supplementary Table 2: Pass rates by release subset and external review status. Rows include every evaluated model configuration. Values are unweighted means of per-problem pass rates after aggregating repeated runs within each problem; uncertainty is reported as the half-width of a 95% hierarchical bootstrap confidence interval from 20,000 resamples, resampling problems and repeated runs within each problem. The public release subset (n=10) reflects the final public case-study set; the Artificial Analysis subset (n=50) follows the final AA reporting set. The review-status columns are restricted to the 129-problem evaluation suite (82 externally reviewed; 47 not externally reviewed).