# ProteomicsBench-Bottom-Up

Benchmarking AI agents on the analysis of mass spectrometry-based bottom-up proteomics data

**Aaron Elias Robinson[1,*], Patrick Tyler Garrett[1,*], Ananya Gupta[1], Asmita Lagwankar[1], Zhe Wang[1], Tamara Varga[1], Yuyi Liu[1], Ian Diks[1], Ramesh Ramasamy[1], Kenny Workman[1]**

[1]LatchBio, San Francisco, CA, USA. [*]These authors contributed equally.
Correspondence: aaron.robinson@latch.bio, patrick.garrett@latch.bio, kenny@latch.bio

### Abstract

Bottom-up proteomics digests proteins into peptides and uses mass spectrometry to identify and quantify peptides, proteins, and their modifications. Interpreting this data involves a series of experiment-specific analytical decisions, with an incorrect choice still producing plausible looking results. Whether AI agents can make these decisions reliably on real data remains an open question. We introduce ProteomicsBench-Bottom-Up, a benchmark of 49 evaluations built from public mass spectrometry-based proteomics datasets. In each evaluation, the agent analyzes data files from a published study to answer a question, and a deterministic grader verifies the answer against the known result. Most questions cannot be answered correctly using a standard processing pipeline, because the agent has to notice something about the experimental design, the acquisition or the limits of the software, as an expert analyst would. Agents ran in offline sandboxes with the data files and open-source proteomics tools, and each of the 38 model–harness configurations (22 models) was run three times on every evaluation. The highest overall pass rate was 49.7% (Gemini 4 Neon / Antigravity), followed by Gemini 4 Neon / PI (41.5%) and Claude Fable 5.2 / Claude Code (40.8%). Of the 22 models, 15 stayed below 10%, and averaged over models, Inference was the hardest stage (5.5%) and Quantification the easiest (19.5%).

## Introduction

Proteins drive most cellular processes, from replicating DNA and metabolizing nutrients to relaying signals between cells. Proteins are also the targets of most FDA-approved drugs[1] and serve as key biomarkers for monitoring health and diagnosing disease[2]. Mass spectrometry identifies and quantifies them at scale by ionizing molecules and measuring their mass-to-charge ratio ($m/z$)[3,4]. There are two main approaches to analyzing proteins through mass spectrometry: top-down, which measures intact proteins[5], and bottom-up, which first digests the proteins into shorter sequences called peptides[6]. Bottom-up is the more widely used technique and is the subject of this work.

An analyst makes many decisions between the raw spectra and a final result. Technical artifacts are pervasive in mass spectrometry runs: batch effects are common[7], instrument performance varies between runs and must be monitored[8], and largely manual sample preparation introduces both technical variability and sporadic handling errors. Quality control is therefore required to determine which runs are suitable for downstream analysis. Database searches must be configured for the experimental conditions and include both the target proteome and common contaminants originating from the reagents or sample handling[9]. The false discovery rate must be controlled[10], and peptide evidence must be assigned to proteins and modification sites[11,12]. Depending on the design of the experiment, the quantities must be normalized[13] and tested with a statistical method that fits that design[14]. Search engines and statistical packages often give no warning when a setting is wrong: a misconfigured search still runs to completion and reports identifications and fold changes that look normal. Even a correctly configured analysis often leaves mislocalized post-translational modifications, misidentified sequences and quantification errors. These are often the hardest errors to find, because they hide among the correct answers.

AI agents can now run biological analysis workflows[15], including proteomics workflows[16–18]. In doing so, they could make proteomics more accessible to the many scientists who are not specialists, opening considerable potential for biological and biomedical research. Whether that potential is realized depends on whether agents make these decisions as an experienced analyst would, and this remains an open question. This work extends our earlier agent benchmarks for single-cell sequencing[19], spatial biology[20,21], preclinical pharmacology[22] and pathogen genomic surveillance[23] to proteomics. Other benchmarks cover general bioinformatics[24] and biology research tasks[25]. None is specific to mass spectrometry proteomics. The closest includes proteomics pipelines as one of several task areas[26]. There is therefore no measure of how well agents perform at each stage of a proteomics analysis, or of whether they catch the errors that the software does not report.

Here we present ProteomicsBench-Bottom-Up, a benchmark of 49 evaluations built from public datasets, to provide one. Each evaluation consists of a prompt with the experimental background and a solvable question, the data files needed to answer it, and a sandbox with the necessary tools and software. The agent performs the analysis and answers the question, and the answer is then graded pass or fail against the ground truth.

## Benchmark Construction

### Evaluations

Each evaluation consists of three components: a prompt, data files, and a grader. The prompt provides the experimental context, asks a scientific question, and specifies the answer format, such as a set of proteins or a numerical value. The data files are publicly available files or reprocessed versions of them, typically taken from a publication's uploaded material. These may include raw instrument files, search results, and quantitative matrices. Where possible, the files were sanitized to prevent the original study from being identified from memory. The grader holds the ground truth and deterministically grades the agent's answer as pass or fail.

In most evaluations, there is some property of the sample or data that the analyst has to notice and account for to answer correctly (35 different properties across the 49 evaluations). These are all things that come up in real proteomics data, but the prompt does not point them out. To get the ground truth, we analyzed the deposited data from each study[27,28] with and without that property

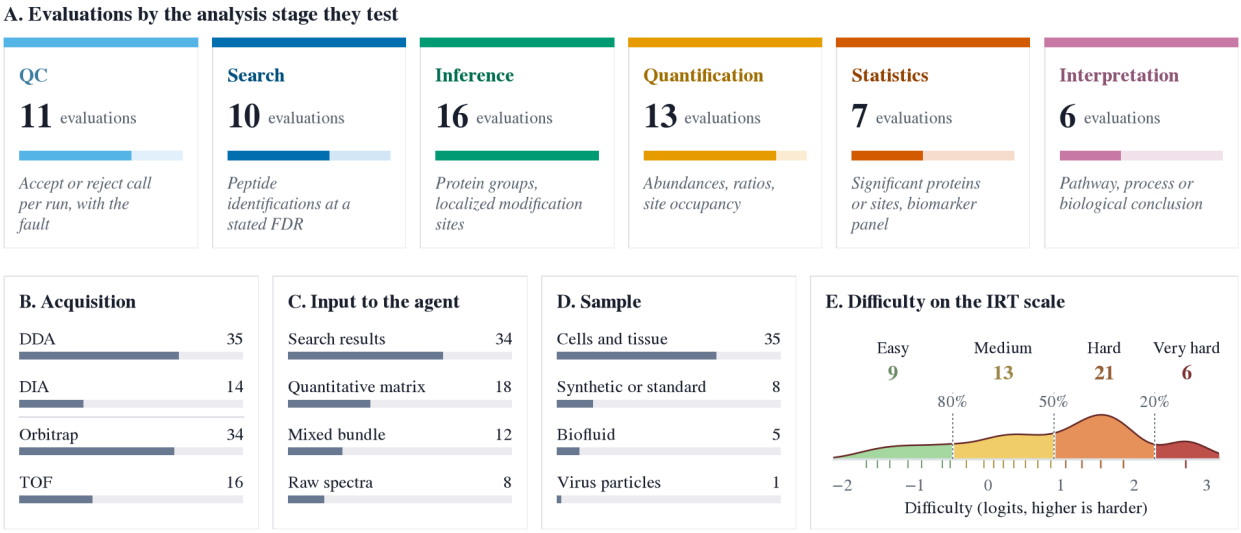



Figure 1: **Benchmark anatomy.** The 49 evaluations, built from 42 public datasets, by (A) the analysis stage they test, (B) acquisition method and mass analyzer, (C) input given to the agent, (D) sample type and (E) difficulty. In (A), each card counts the evaluations that test the stage, as in Figure 3, on a bar scale shared across cards, and names what the agent reports at that stage. An evaluation counts once for each stage and input type it has. In (B)-(D), each bar sits on a pale bar whose full length is all 49 evaluations, and evaluations with no recorded value for a category are left out. (E) Difficulty of each evaluation from a one-parameter item response theory (IRT, Rasch) model fitted on the pass rates of all 22 models, in logits, shown as a density with one tick per evaluation. Tiers are cut where the strongest model's expected chance of passing falls to 80%, 50% and 20% (dashed lines): Easy, Medium, Hard and Very hard, with the count of each above its band. DDA is data-dependent acquisition, DIA is data-independent acquisition and TOF is time of flight.

accounted for. The observed difference between the two answers forms the basis of the grader's tolerance bands.

For example, one evaluation asks how many deamidated sites are present in a dataset. Deamidation adds 0.984 Da to a peptide, but in this dataset the instrument also picked the carbon-13 isotope peak instead of the monoisotopic peak for some peptides, which makes those peptides look 1.003 Da heavier than they really are[29]. These mass shifts are only 0.019 Da apart, so a search with a wide mass tolerance and no isotope correction can misassign this modification[30]. An experienced analyst would catch this because the false matches all show the same precursor mass error of 0.019 Da.

The evaluations span the six stages of a bottom-up analysis (Figure 1): quality control and triage (QC), search and identification (Search), protein inference and modification-site localization (Inference), quantification and normalization (Quantification), statistics and biomarker discovery (Statistics), and biological interpretation (Interpretation). An evaluation can test more than one stage, so the stage counts sum to more than the 49 evaluations. In 35 evaluations, the spectra were acquired by data-dependent acquisition, and in 14 by data-independent acquisition[31].

Each evaluation went through several rounds of review before it was included in the benchmark. These included automated checks of the prompt, grader, data files and agent trajectories. These checks tested whether the answer could be guessed from the prompt or reached without opening the data files. They also tested the grader in both directions: it had to reject the answer from the default analysis (no false positives) and accept a correct answer reached by another valid route (no false negatives). Each evaluation was also reviewed by three individuals. Lastly, a proteomics expert attempted 16 of the 49 evaluations with the same files and tools as the agents and solved every one, including all 5 evaluations that no agent passed.

### Models and runs

We tested 22 models in 5 harnesses (PI, Claude Code, Codex, Antigravity and Grok Build), where the harness is the program that runs the model, executes its tool calls and manages its context[32]. We refer to each model–harness pair as a configuration, and there were 38 of these in total. Each configuration was run three times on every evaluation, with each run in a new sandbox holding a Python environment and open-source proteomics software, including Sage[33], Comet[34], AlphaDIA[35], Casanovo[36] and msconvert[37]. The sandboxes were offline, so an agent could not search the literature or download the original study's results. Each run stopped when the agent submitted its answer as a JSON object or after 6 hours. The grader scored every field of the answer, and a run passed only if all of them passed. Anything else, including a refusal, an error or an answer that could not be parsed, was scored as a fail. A configuration's pass rate is the fraction of its three runs that passed on each evaluation, averaged across evaluations so that each counts equally. The runs took place from November 21 to 23, 2026. Of the 5,586 runs, 109 ended in an error, 79 of them at the time limit, and 18 were refusals.

## Results

### Leaderboard

Two models led the benchmark, and they could not be separated statistically. Gemini 4 Neon / Antigravity had the highest pass rate at 49.7%, passing 73 of its 147 runs (Figure 2 A), followed by Gemini 4 Neon / PI at 41.5% and Claude Fable 5.2 / Claude Code at 40.8%. A result given for a model, and not for a configuration, is that of the model's best configuration. On that basis, Gemini 4 Neon was 9 points ahead of Claude Fable 5.2, a difference that was not statistically significant (paired 95% confidence interval −6 to 24 points).

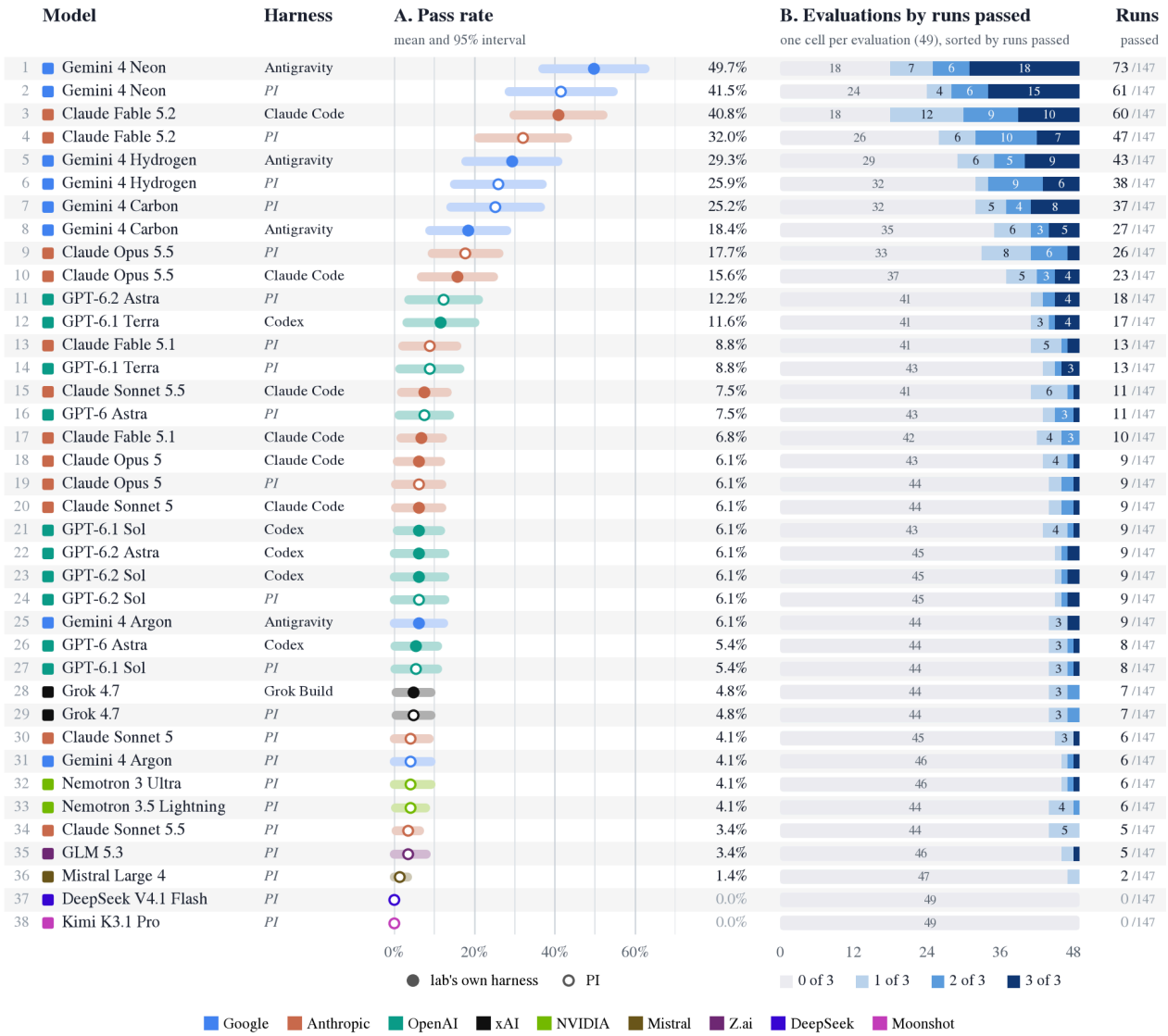



Figure 2: **Pass rate and repeatability by configuration.** (A) Mean pass rate over the 49 evaluations, with 95% Student's *t* intervals over per-evaluation pass rates, cut off at 0%. Color marks the model's lab. Filled points mark the lab's own harness, and open points mark the PI harness. Configurations are ranked by mean pass rate, and Figure 4 reuses these ranks. (B) The 49 evaluations of each configuration by how many of the three runs passed, from none (lightest) to all three (darkest). Numbers are evaluation counts, and the column on the right gives the runs passed out of the 147 attempted. Refusals and errors count as failures. Differences between two configurations reported in the text use paired *t* intervals over the per-evaluation differences.

Most models passed few runs. The mean pass rate across all 38 configurations was 11.9%, with a median of 6.1%. The third-ranked model, Gemini 4 Hydrogen, had a pass rate of 29.3%. Of the 22 models, 15 were below 10%, and 2 of them passed no run in either harness.

The harness changed the pass rate only for the strongest models (Figure S1). Of the 16 models tested in two harnesses, 3 differed by more than run-to-run noise: Claude Fable 5.2, Gemini 4 Neon and Gemini 4 Carbon. Claude Fable 5.2 had a pass rate of 40.8% in Claude Code and 32.0% in PI. For Gemini 4 Neon, 12 of the 147 runs in PI ended in an error, against 1 in Antigravity, so most of its additional failures in PI were runs that did not finish rather than wrong answers.

## Performance across analysis stages

Protein inference and modification-site localization was the hardest stage. Averaged over all 22 models, the pass rate on the 16 Inference evaluations was 5.5%, against 8.6% to 19.5% at the other five stages (Figure 3 A). The two leading models were also weak there: Gemini 4 Neon passed 27.1% of its Inference runs and Claude Fable 5.2 passed 29.2%, against 76.2% and 66.7% on the 7 Statistics evaluations. Single evaluations also differed widely in difficulty: averaged across models, the pass rate ranged from 0% on the hardest evaluation to 43.9% on the easiest (Figure 3 B).

## Repeatability and agreement between models

A configuration that passed an evaluation once often failed it in another run. We ran each configuration three times on every evaluation so that this could be measured[38]. Most configuration–evaluation pairs (81.7%) failed in all three runs. Of the 341 pairs

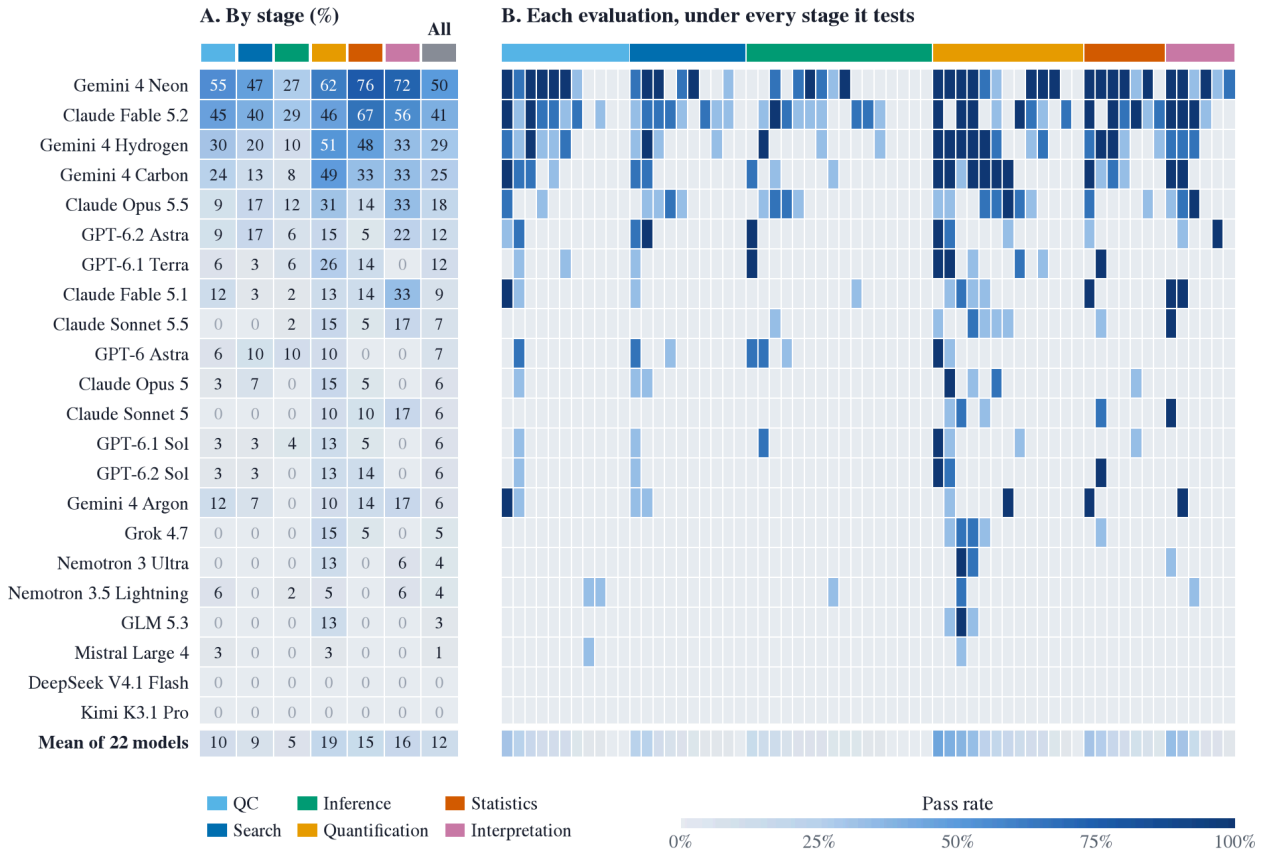



Figure 3: **Pass rate by model, analysis stage and evaluation.** One row per model, each shown in its best configuration from Figure 2 and ordered by that pass rate. The last row is the mean of the models. (A) Pass rate in percent on the evaluations that test each stage, with all 49 evaluations in the last column. (B) One column per evaluation under each stage it tests, colored by pass rate on the scale of (A), ordered within a stage from the most to the least passed. An evaluation that tests more than one stage counts in each of them in (A) and appears under each of them in (B). Evaluation names are withheld. Refusals and errors count as failures.

that passed at least once, only 117 (34.3%) passed in all three runs (Figure 2 B). The two leading configurations showed the same pattern: Gemini 4 Neon / Antigravity passed 63.3% of evaluations at least once but only 36.7% in all three runs, and Gemini 4 Neon / PI passed 51.0% and 30.6%.

The two leading models passed different evaluations (Figure 3 B). Gemini 4 Neon passed 7 evaluations in at least half of its runs that Claude Fable 5.2 never passed, and Claude Fable 5.2 passed 6 in at least half of its runs that Gemini 4 Neon never passed. Per-evaluation pass rates agreed closely between the two harnesses of one model (Spearman $\rho = 0.86$ for Gemini 4 Neon and 0.73 for Claude Fable 5.2). Between the two models they agreed only weakly ($\rho = 0.23$, 95% confidence interval −0.05 to 0.48).

Combining the two leading models therefore raises the pass rate. Taking the better of the two on each evaluation gives 64.6%, against 49.7% and 40.8% for each alone, and the next-best pair of models reaches 57.1%. These rates are optimistic, because they assume it is known which model to trust on each evaluation, and because the best of two results also benefits from chance.

### Cost and time

A median run took 9 minutes and 1.3 million tokens and cost $1.00. All 5,586 runs together, including those that did not complete, cost $17,085. Configurations differed widely (Figure 4): Claude Fable 5.2 / PI was the most expensive at a median of $13.89 per run, and Nemotron 3.5 Lightning / PI the least expensive at $0.01.

More expensive configurations passed more often, and no inexpensive configuration reached a useful pass rate. Pass rate rose with the median cost of a run across the 38 configurations (Spearman $\rho = 0.68$). Of the 17 configurations with a median cost under $1 per run, the best had a pass rate of 12.2%. Among the leaders, cost did not follow pass rate: Gemini 4 Neon / Antigravity had 49.7% at $5.25 per run, and Claude Fable 5.2 / Claude Code had 40.8% at $12.23.

## Discussion

Current AI agents are not ready for autonomous proteomics analysis. The strongest configuration had a pass rate of 49.7%, and 15 of 22 models were below 10%. Pass rates were lowest in the first three stages of an analysis: averaged over models, they were 10.3% for quality control, 8.6% for search and identification and 5.5% for protein inference and modification-site localization.

Running a standard analysis was not the problem. On clean datasets with a straightforward analysis, agents generally followed standard pipelines, configured searches correctly and returned sensible answers. The evaluations we built on such data did not differentiate the models. Data this clean are not the norm[7–9]. In most evaluations in the benchmark, all built from published

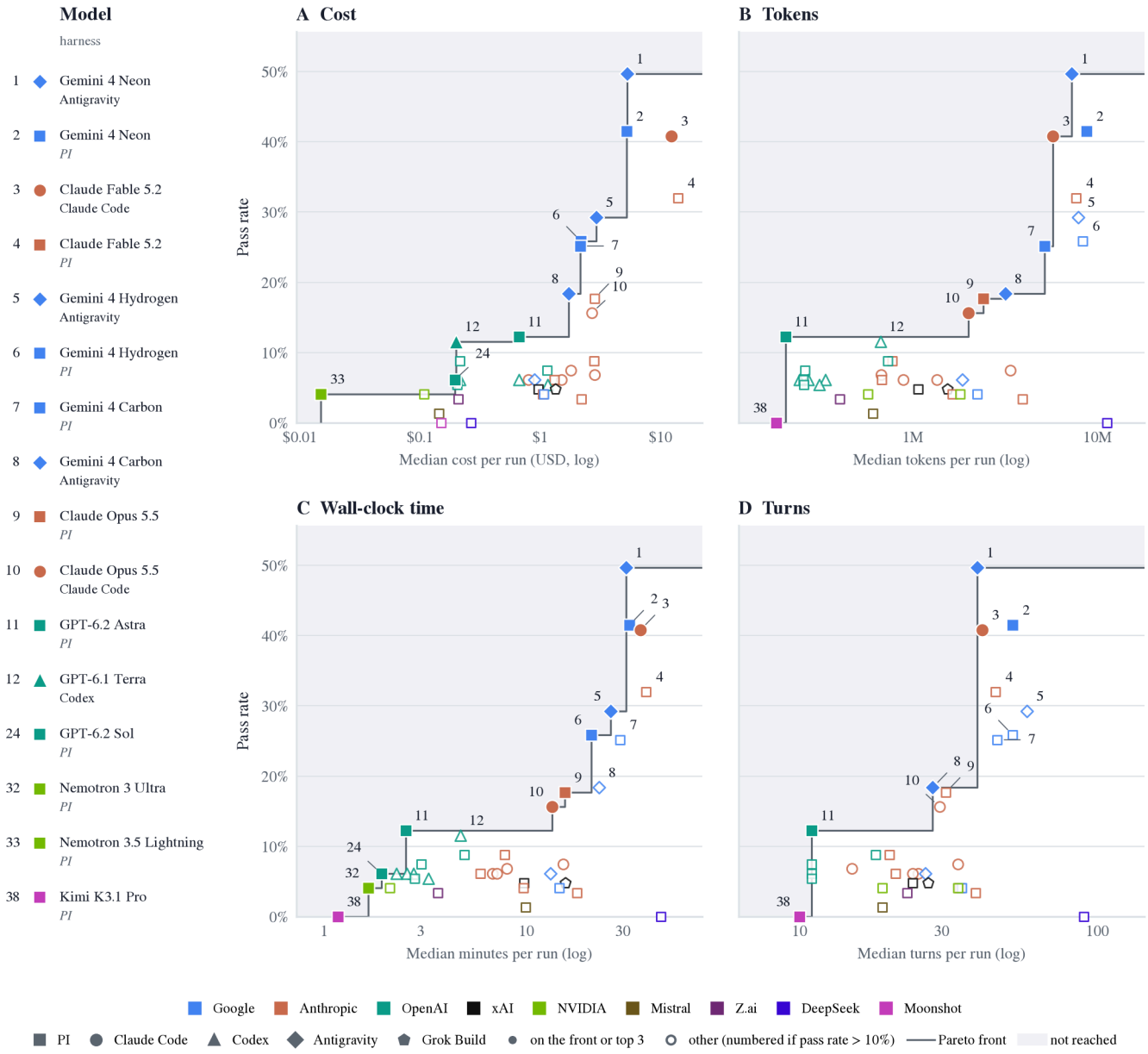



Figure 4: **Pass rate against the cost, tokens, wall-clock time and turns of a run.** Each point is one configuration: its pass rate over the 49 evaluations against the median (A) cost, (B) tokens, (C) wall-clock minutes and (D) turns of its completed runs, on logarithmic axes. Cost is the amount the harness reported, and tokens are the sum of input, cached input and output tokens. Color marks the model's lab and shape the harness. The solid line is the Pareto front, the best pass rate reached at that value or less, and no configuration reached the shaded region above it. Configurations on the front and the top three are filled, and the others are hollow. Filled points, and every configuration with a pass rate above 10%, are labeled with their rank from Figure 2, with a thin leader line where markers overlap. The key on the left names each labeled configuration. Codex records one turn for every run, so its configurations are absent from (D). Table S1 lists the plotted values.

studies, the analyst has to notice and account for a property of the sample or data to answer correctly. A published dataset is one its authors judged good enough to publish. Routine laboratory data has had no such selection, so we expect such properties to be more common there. An agent that missed such a property still completed the analysis and returned an answer in the requested format, without reporting a problem. A non-specialist cannot tell such an answer from a correct one, so a proteomics expert is still needed to check the analysis.

For researchers using agents to do or help with proteomic data analysis, our findings point to two directions. First, combining models is a promising near-term strategy to improve reliability. Gemini 4 Neon and Claude Fable 5.2 achieved similar overall pass rates, yet they correlated weakly across evaluations and passed different ones. Taking the better of the two on each evaluation raises the pass rate to at best 64.6%. Second, the harness mattered for the strongest models. The 3 models whose pass rates differed between harnesses by more than run-to-run noise (Claude Fable 5.2, Gemini 4 Neon and Gemini 4 Carbon) are all among the four highest-ranked, and most of the other models had pass rates too

low in both harnesses for a difference to show. All 5 harnesses we tested are general-purpose. A promising next step is to equip one with proteomics context, such as routine checks for sample quality, isotope misassignment and contaminant proteins, and test whether this raises pass rates.

The main limitation of this benchmark is its restriction to open-source software. Many published proteomics studies rely on commercial or academic-license tools, which were out of the scope of this analysis. When possible we provided the best-known open-source equivalents. Evaluations were designed so that the conclusion does not depend on the tool. Our ground truth also comes from our own reanalysis of the deposited data rather than from the original publications. This was deliberate: language models have read many of these papers, and an answer copied from a paper could be recalled rather than derived from the data. Finally, this release covers only bottom-up proteomics. Top-down and imaging remain open areas for future mass spectrometry-based proteomics benchmarks.

## Data availability

The evaluations, ground truth and graders of the benchmark set are withheld so that models cannot be trained on them[39,40], which keeps the benchmark measuring analysis rather than recall. Three example evaluations, which are not part of the benchmark set, are publicly released to show the structure of an evaluation and its grader.

## References


(1) Santos, R.; Ursu, O.; Gaulton, A.; Bento, A. P.; Donadi, R. S.; Bologa, C. G.; Karlsson, A.; Al-Lazikani, B.; Hersey, A.; Oprea, T. I.; Overington, J. P. A Comprehensive Map of Molecular Drug Targets. *Nature Reviews Drug Discovery* **2017**, *16* (1), 19–34. https://doi.org/10.1038/nrd.2016.230.

(2) Geyer, P. E.; Holdt, L. M.; Teupser, D.; Mann, M. Revisiting Biomarker Discovery by Plasma Proteomics. *Molecular Systems Biology* **2017**, *13* (9), 942. https://doi.org/10.15252/msb.20156297.

(3) Aebersold, R.; Mann, M. Mass Spectrometry-Based Proteomics. *Nature* **2003**, *422* (6928), 198–207. https://doi.org/10.1038/nature01511.

(4) Aebersold, R.; Mann, M. Mass-Spectrometric Exploration of Proteome Structure and Function. *Nature* **2016**, *537* (7620), 347–355. https://doi.org/10.1038/nature19949.

(5) Toby, T. K.; Fornelli, L.; Kelleher, N. L. Progress in Top-down Proteomics and the Analysis of Proteoforms. *Annual Review of Analytical Chemistry* **2016**, *9* (1), 499–519. https://doi.org/10.1146/annurev-anchem-071015-041550.

(6) Zhang, Y.; Fonslow, B. R.; Shan, B.; Baek, M.-C.; Yates, J. R. Protein Analysis by Shotgun/bottom-up Proteomics. *Chemical Reviews* **2013**, *113* (4), 2343–2394. https://doi.org/10.1021/cr3003533.

(7) Čuklina, J.; Lee, C. H.; Williams, E. G.; Sajic, T.; Collins, B. C.; Rodríguez Martínez, M.; Sharma, V. S.; Wendt, F.; Goetze, S.; Keele, G. R.; Wollscheid, B.; Aebersold, R.; Pedrioli, P. G. A. Diagnostics and Correction of Batch Effects in Large-Scale Proteomic Studies: A Tutorial. *Molecular Systems Biology* **2021**, *17* (8), e10240. https://doi.org/10.15252/msb.202110240.

(8) Bittremieux, W.; Valkenborg, D.; Martens, L.; Laukens, K. Computational Quality Control Tools for Mass Spectrometry Proteomics. *Proteomics* **2017**, *17* (3–4), 1600159. https://doi.org/10.1002/pmic.201600159.

(9) Frankenfield, A. M.; Ni, J.; Ahmed, M.; Hao, L. Protein Contaminants Matter: Building Universal Protein Contaminant Libraries for DDA and DIA Proteomics. *Journal of Proteome Research* **2022**, *21* (9), 2104–2113. https://doi.org/10.1021/acs.jproteome.2c00145.

(10) Elias, J. E.; Gygi, S. P. Target-Decoy Search Strategy for Increased Confidence in Large-Scale Protein Identifications by Mass Spectrometry. *Nature Methods* **2007**, *4* (3), 207–214. https://doi.org/10.1038/nmeth1019.

(11) Nesvizhskii, A. I.; Aebersold, R. Interpretation of Shotgun Proteomic Data: The Protein Inference Problem. *Molecular & Cellular Proteomics* **2005**, *4* (10), 1419–1440. https://doi.org/10.1074/mcp.r500012-mcp200.

(12) Beausoleil, S. A.; Villén, J.; Gerber, S. A.; Rush, J.; Gygi, S. P. A Probability-Based Approach for High-Throughput Protein Phosphorylation Analysis and Site Localization. *Nature Biotechnology* **2006**, *24* (10), 1285–1292. https://doi.org/10.1038/nbt1240.

(13) Cox, J.; Hein, M. Y.; Luber, C. A.; Paron, I.; Nagaraj, N.; Mann, M. Accurate Proteome-Wide Label-Free Quantification by Delayed Normalization and Maximal Peptide Ratio Extraction, Termed MaxLFQ. *Molecular & Cellular Proteomics* **2014**, *13* (9), 2513–2526. https://doi.org/10.1074/mcp.m113.031591.

(14) Choi, M.; Chang, C.-Y.; Clough, T.; Broudy, D.; Killeen, T.; MacLean, B.; Vitek, O. MSstats: An R Package for Statistical Analysis of Quantitative Mass Spectrometry-Based Proteomic Experiments. *Bioinformatics* **2014**, *30* (17), 2524–2526. https://doi.org/10.1093/bioinformatics/btu305.

(15) Huang, K.; Zhang, S.; Wang, H.; Qu, Y.; Lu, Y.; Li, R.; Roohani, Y.; Qiu, L.; Cao, S.; Li, G.; Zhang, J.; Yin, D.; Wierenga, R.; Kavi, D.; Liu, S.; She, T.; Marwaha, S.; Carter, J. N.; Zhou, X.; Wheeler, M. T.; Bernstein, J. A.; Wang, M.; He, P.; Zhou, J.; Snyder, M. P.; Cong, L.; Regev, A.; Leskovec, J. Autonomous Biomedical Research with an Artificial Intelligence Agent. *Science* **2026**, *393* (6813), eadz4351. https://doi.org/10.1126/science.adz4351.

(16) Ding, N.; Qu, S.; Xie, L.; Li, Y.; Liu, Z.; Zhang, K.; Xiong, Y.; Zuo, Y.; Chen, Z.; Hua, E.; Lv, X.; Sun, Y.; Li, Y.; Li, D.; He, F.; Zhou, B. *Automating Exploratory Proteomics Research via Language Models*. https://arxiv.org/abs/2411.03743.

(17) Li, J.; Charkow, J.; Gao, M.; Li, J.; Röst, H. ChatDIA: A Zero-Shot Large Language Model Workflow for Targeted Analysis of Data-Independent Acquisition Mass Spectrometry Data, 2026. https://doi.org/10.64898/2026.02.11.705360.

(18) Liao, Y.; Cheng, J.; Xiao, Z.; Yi, T.; Ding, J.; Zhang, W. *MSRyzen: An Agentic Platform for Unified Proteomics Analysis and Evidence-Centric Visualization*. https://arxiv.org/abs/2611.27246.

(19) Workman, K.; Yang, Z.; Muralidharan, H.; Abdulali, A.; Le, H. *scBench: Evaluating AI Agents on Single-Cell RNA-seq Analysis*. https://arxiv.org/abs/2602.09063.

(20) Workman, K.; Yang, Z.; Muralidharan, H.; Le, H. *SpatialBench: Can Agents Analyze Real-World Spatial Biology Data?*. https://arxiv.org/abs/2512.21907.

(21) Diks, I.; Muralidharan, H.; Proctor, T.; Workman, K. *Verifiable Benchmarking of Long-Horizon Spatial Biology*. https://arxiv.org/abs/2605.28065.

(22) Le, H.; Ramasamy, R.; Urrutia, A.; Yazdani, M.; Proctor, T.; Workman, K. *TxBench-PP: Analyzing AI Agent Performance on Small-Molecule Preclinical Pharmacology*. https://arxiv.org/abs/2606.19245.

(23) Bhasin, H.; Flyangolts, K.; Wang, D.; Seeyave, E.; Banerjee, A.; Darling, A.; Stallings, J.; Stern, D.; Higdon, S.; Duvallet, C.; Tegomoh, B.; Workman, K. *BioSecBench-Surveillance: A Verifiable Benchmark for AI Agents in Pathogen Genomic Surveillance*. https://arxiv.org/abs/2607.19262.

(24) Mitchener, L.; Laurent, J. M.; Andonian, A.; Tenmann, B.; Narayanan, S.; Wellawatte, G. P.; White, A.; Sani, L.; Rodriques, S. G. *BixBench: a Comprehensive Benchmark for LLM-based Agents in Computational Biology*. https://arxiv.org/abs/2503.00096.

(25) Laurent, J. M.; Janizek, J. D.; Ruzo, M.; Hinks, M. M.; Hammerling, M. J.; Narayanan, S.; Ponnapati, M.; White, A. D.; Rodriques, S. G. LAB-Bench: Measuring Capabilities of Language Models for Biology Research, 2024. https://doi.org/10.48550/arXiv.2407.10362.

(26) Guo, W.; Zhang, M.; Han, B.; Ma, Y.; Leng, Y.; Hebbar, S.; Zhou, X.; Gu, W.; Yang, X.; Dhar, S. PromptBio-Bench: Benchmarking LLM-Based Bioinformatics Agents for End-to-End Data Analysis, 2026. https://doi.org/10.64898/2026.05.05.723092.

(27) Perez-Riverol, Y.; Bandla, C.; Kundu, D. J.; Kamatchinathan, S.; Bai, J.; Hewapathirana, S.; John, N. S.; Prakash, A.; Walzer, M.; Wang, S.; Vizcaíno, J. A. The PRIDE Database at 20 Years: 2025 Update. *Nucleic Acids Research* **2025**, *53* (D1), D543–D553. https://doi.org/10.1093/nar/gkae1011.

(28) Deutsch, E. W.; Bandeira, N.; Perez-Riverol, Y.; Sharma, V.; Carver, J. J.; Mendoza, L.; Kundu, D. J.; Wang, S.; Bandla, C.; Kamatchinathan, S.; Hewapathirana, S.; Pullman, B. S.; Wertz, J.; Sun, Z.; Kawano, S.; Okuda, S.; Watanabe, Y.; MacLean, B.; MacCoss, M. J.; Zhu, Y.; Ishihama, Y.; Vizcaíno, J. A. The ProteomeXchange Consortium at 10 Years: 2023 Update. *Nucleic Acids Research* **2023**, *51* (D1), D1539–D1548. https://doi.org/10.1093/nar/gkac1040.

(29) Chick, J. M.; Kolippakkam, D.; Nusinow, D. P.; Zhai, B.; Rad, R.; Huttlin, E. L.; Gygi, S. P. A Mass-Tolerant Database Search Identifies a Large Proportion of Unassigned Spectra in Shotgun Proteomics as Modified Peptides. *Nature Biotechnology* **2015**, *33* (7), 743–749. https://doi.org/10.1038/nbt.3267.

(30) Hao, P.; Ren, Y.; Alpert, A. J.; Sze, S. K. Detection, Evaluation and Minimization of Nonenzymatic Deamidation in Proteomic Sample Preparation. *Molecular & Cellular Proteomics* **2011**, *10* (10), O111.009381. https://doi.org/10.1074/mcp.O111.009381.

(31) Gillet, L. C.; Navarro, P.; Tate, S.; Röst, H.; Selevsek, N.; Reiter, L.; Bonner, R.; Aebersold, R. Targeted Data Extraction of the MS/MS Spectra Generated by Data-Independent Acquisition: A New Concept for Consistent and Accurate Proteome Analysis. *Molecular & Cellular Proteomics* **2012**, *11* (6), O111.016717. https://doi.org/10.1074/mcp.o111.016717.

(32) Yang, J.; Jimenez, C.; Wettig, A.; Lieret, K.; Yao, S.; Narasimhan, K.; Press, O. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. In *Advances in Neural Information Processing Systems 37*; 2024; pp 50528–50652. https://doi.org/10.52202/079017-1601.

(33) Lazear, M. R. Sage: An Open-Source Tool for Fast Proteomics Searching and Quantification at Scale. *Journal of Proteome Research* **2023**, *22* (11), 3652–3659. https://doi.org/10.1021/acs.jproteome.3c00486.

(34) Eng, J. K.; Jahan, T. A.; Hoopmann, M. R. Comet: An Open-Source MS/MS Sequence Database Search Tool. *Proteomics* **2013**, *13* (1), 22–24. https://doi.org/10.1002/pmic.201200439.

(35) Wallmann, G.; Skowronek, P.; Brennsteiner, V.; Lebedev, M.; Thielert, M.; Steigerwald, S.; Kotb, M.; Despard, O.; Heymann, T.; Zhou, X.-X.; Strauss, M. T.; Ammar, C.; Willems, S.; Schwörer, M.; Zeng, W.-F.; Mann, M. AlphaDIA Enables DIA Transfer Learning for Feature-Free Proteomics. *Nature Biotechnology* **2026**, *44* (7), 1168–1177. https://doi.org/10.1038/s41587-025-02791-w.

(36) Yilmaz, M.; Fondrie, W. E.; Bittremieux, W.; Melendez, C. F.; Nelson, R.; Ananth, V.; Oh, S.; Noble, W. S. Sequence-to-Sequence Translation from Mass Spectra to Peptides with a Transformer Model. *Nature Communications* **2024**, *15*, 6427. https://doi.org/10.1038/s41467-024-49731-x.

(37) Chambers, M. C.; Maclean, B.; Burke, R.; Amodei, D.; Ruderman, D. L.; Neumann, S.; Gatto, L.; Fischer, B.; Pratt, B.; Egertson, J.; Hoff, K.; Kessner, D.; Tasman, N.; Shulman, N.; Frewen, B.; Baker, T. A.; Brusniak, M.-Y.; Paulse, C.; Creasy, D.; Flashner, L.; Kani, K.; Moulding, C.; Seymour, S. L.; Nuwaysir, L. M.; Lefebvre, B.; Kuhlmann, F.; Roark, J.; Paape, R.; Suckau, D.; Hemenway, T.; Huhmer, A.; Langridge, J.; Connolly, B.; Chadick, T.; Holly, K.; Eckels, J.; Deutsch, E. W.; Moritz, R. L.; Katz, J. E.; Agus, D. B.; MacCoss, M.; Tabb, D. L.; Mallick, P. A Cross-Platform Toolkit for Mass Spectrometry and Proteomics. *Nature Biotechnology* **2012**, *30* (10), 918–920. https://doi.org/10.1038/nbt.2377.

(38) Yao, S.; Shinn, N.; Razavi, P.; Narasimhan, K. \tau-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. In *International Conference on Learning Representations (ICLR 2025)*; 2025; pp 74824–74876.

(39) Sainz, O.; Campos, J.; García-Ferrero, I.; Etxaniz, J.; Lopez de Lacalle, O.; Agirre, E. NLP Evaluation in Trouble: On the Need to Measure LLM Data Contamination for Each Benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*; 2023; pp 10776–10787. https://doi.org/10.18653/v1/2023.findings-emnlp.722.

(40) Jacovi, A.; Caciularu, A.; Goldman, O.; Goldberg, Y. Stop Uploading Test Data in Plain Text: Practical Strategies for Mitigating Data Contamination by Evaluation Benchmarks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*; 2023; pp 5075–5084. https://doi.org/10.18653/v1/2023.emnlp-main.308.

# Supporting Information

## ProteomicsBench-Bottom-Up

*Aaron Elias Robinson, Patrick Tyler Garrett, Ananya Gupta, Asmita Lagwankar, Zhe Wang, Tamara Varga, Yuyi Liu, Ian Diks, Ramesh Ramasamy, Kenny Workman*

Figure S1 compares the pass rate of each model in the PI harness with its pass rate in the lab's own harness.

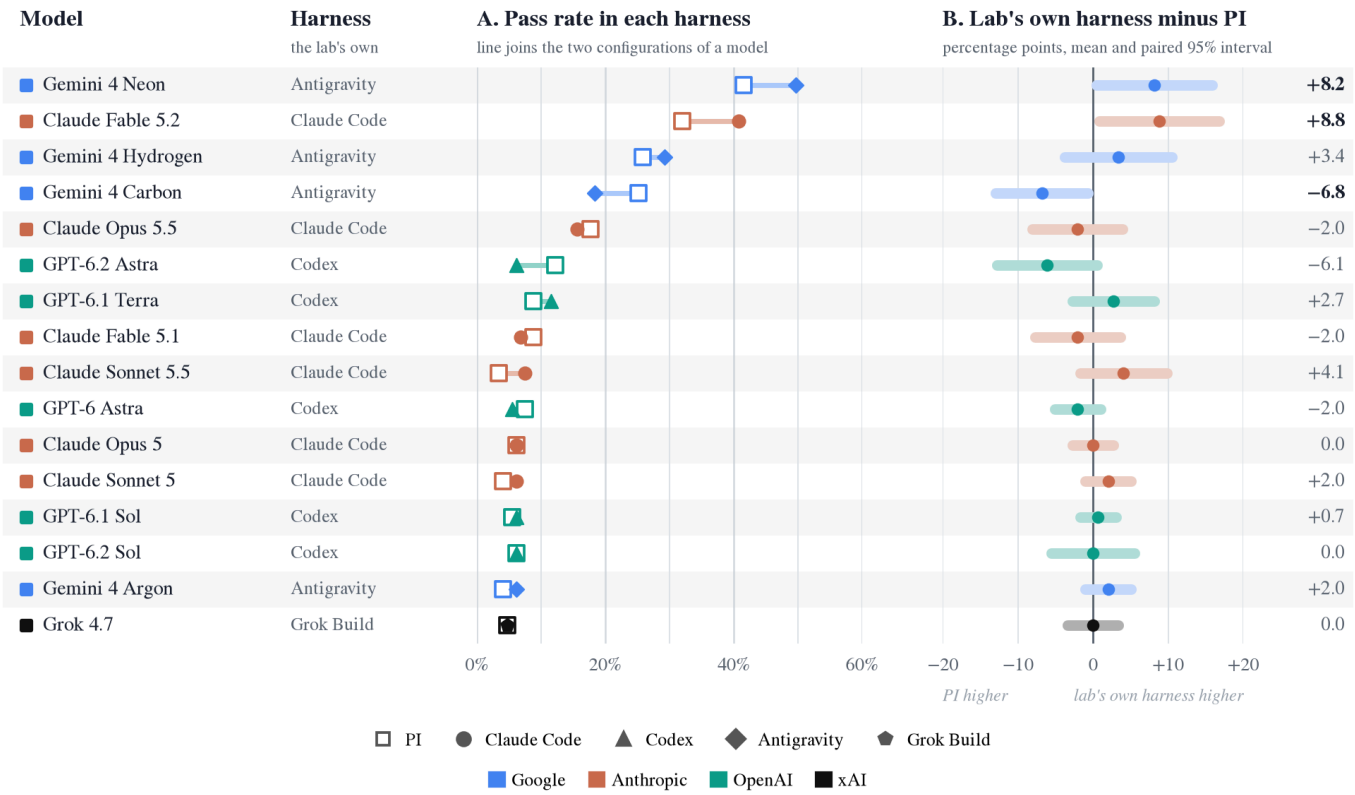



Figure S1: **Effect of the harness on pass rate.** (A) Pass rate over the 49 evaluations of the 16 models tested in two harnesses. A line joins the two configurations of a model. Open squares mark the PI harness, and filled shapes mark the lab's own harness. Color marks the model's lab. Models are ordered by the pass rate of their best configuration, as in Figure 2. (B) Pass rate in the lab's own harness minus pass rate in the PI harness, in percentage points: the mean of the per-evaluation differences with its 95% paired Student's *t* interval. A value in bold has an interval that excludes zero. Refusals and errors count as failures.

Table S1 lists every configuration with the values that Figure 2 and Figure 4 plot.

Table S1: **Pass rate, runs and cost of every configuration.** One row per configuration, in the rank order of Figure 2, which Figure 4 prints beside its points. Pass rate is the mean over the 49 evaluations, with its 95% Student's *t* interval over per-evaluation pass rates, cut off at 0%. Runs are counted out of the 147 attempted, and errored and refused runs count as failures. Evaluations passed gives how many of the 49 evaluations passed in at least one run and in all three runs. The last four columns are the medians over completed runs that Figure 4 plots: cost in US dollars as the harness reported it, tokens in millions (input, cached input and output), wall-clock minutes and turns. Codex records one turn for every run, so its configurations have no turn count.

| | | | Pass rate (%) | | Runs | | | Evaluations passed | | Median of a completed run | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rank | Model | Harness | Mean | 95% interval | Passed | Errored | Refused | In any run | In all runs | Cost (USD) | Tokens (M) | Minutes | Turns |
| 1 | Gemini 4 Neon | Antigravity | 49.7 | 37.0–62.3 | 73 | 1 | 0 | 31 | 18 | 5.25 | 7.27 | 31.1 | 39.5 |
| 2 | Gemini 4 Neon | PI | 41.5 | 28.6–54.4 | 61 | 12 | 0 | 25 | 15 | 5.19 | 8.73 | 32.3 | 52 |
| 3 | Claude Fable 5.2 | Claude Code | 40.8 | 29.7–51.9 | 60 | 3 | 0 | 31 | 10 | 12.23 | 5.72 | 36.7 | 41 |
| 4 | Claude Fable 5.2 | PI | 32.0 | 20.9–43.0 | 47 | 7 | 0 | 23 | 7 | 13.89 | 7.69 | 39.1 | 45.5 |
| 5 | Gemini 4 Hydrogen | Antigravity | 29.3 | 17.8–40.8 | 43 | 9 | 0 | 20 | 9 | 2.92 | 7.87 | 26.2 | 58 |
| 6 | Gemini 4 Hydrogen | PI | 25.9 | 14.9–36.8 | 38 | 5 | 1 | 17 | 6 | 2.17 | 8.35 | 21.0 | 52 |
| 7 | Gemini 4 Carbon | PI | 25.2 | 14.0–36.3 | 37 | 8 | 1 | 17 | 8 | 2.14 | 5.17 | 29.1 | 46 |
| 8 | Gemini 4 Carbon | Antigravity | 18.4 | 8.8–28.0 | 27 | 4 | 0 | 14 | 5 | 1.72 | 3.17 | 22.9 | 28 |
| 9 | Claude Opus 5.5 | PI | 17.7 | 9.4–26.0 | 26 | 2 | 0 | 16 | 2 | 2.82 | 2.41 | 15.5 | 31 |
| 10 | Claude Opus 5.5 | Claude Code | 15.6 | 6.7–24.6 | 23 | 1 | 3 | 12 | 4 | 2.67 | 2.00 | 13.4 | 29.5 |
| 11 | GPT-6.2 Astra | PI | 12.2 | 3.6–20.9 | 18 | 0 | 0 | 8 | 4 | 0.66 | 0.21 | 2.5 | 11 |
| 12 | GPT-6.1 Terra | Codex | 11.6 | 3.1–20.0 | 17 | 0 | 0 | 8 | 4 | 0.20 | 0.67 | 4.7 | – |
| 13 | Claude Fable 5.1 | PI | 8.8 | 2.1–15.6 | 13 | 2 | 6 | 8 | 2 | 2.78 | 0.77 | 7.8 | 20 |
| 14 | GPT-6.1 Terra | PI | 8.8 | 1.3–16.4 | 13 | 0 | 0 | 6 | 3 | 0.21 | 0.73 | 4.9 | 18 |
| 15 | Claude Sonnet 5.5 | Claude Code | 7.5 | 1.9–13.1 | 11 | 2 | 0 | 8 | 1 | 1.79 | 3.38 | 15.2 | 34 |
| 16 | GPT-6 Astra | PI | 7.5 | 1.2–13.7 | 11 | 0 | 0 | 6 | 1 | 1.13 | 0.26 | 3.0 | 11 |
| 17 | Claude Fable 5.1 | Claude Code | 6.8 | 1.6–12.0 | 10 | 1 | 6 | 7 | 0 | 2.83 | 0.67 | 8.0 | 15 |
| 18 | Claude Opus 5 | Claude Code | 6.1 | 0.7–11.5 | 9 | 1 | 0 | 6 | 1 | 1.50 | 0.89 | 6.8 | 25 |
| 19 | Claude Opus 5 | PI | 6.1 | 0.4–11.9 | 9 | 0 | 1 | 5 | 1 | 1.30 | 0.68 | 5.9 | 21 |
| 20 | Claude Sonnet 5 | Claude Code | 6.1 | 0.4–11.9 | 9 | 4 | 0 | 5 | 1 | 0.79 | 1.35 | 7.2 | 24 |
| 21 | GPT-6.1 Sol | Codex | 6.1 | 0.7–11.5 | 9 | 0 | 0 | 6 | 1 | 0.21 | 0.34 | 2.8 | – |
| 22 | GPT-6.2 Astra | Codex | 6.1 | 0.0–12.5 | 9 | 0 | 0 | 4 | 2 | 0.66 | 0.24 | 2.6 | – |
| 23 | GPT-6.2 Sol | Codex | 6.1 | 0.0–12.5 | 9 | 1 | 0 | 4 | 2 | 0.20 | 0.27 | 2.3 | – |
| 24 | GPT-6.2 Sol | PI | 6.1 | 0.0–12.5 | 9 | 0 | 0 | 4 | 2 | 0.19 | 0.26 | 1.9 | 11 |
| 25 | Gemini 4 Argon | Antigravity | 6.1 | 0.0–12.2 | 9 | 1 | 0 | 5 | 2 | 0.89 | 1.85 | 13.2 | 26.5 |
| 26 | GPT-6 Astra | Codex | 5.4 | 0.1–10.7 | 8 | 0 | 0 | 5 | 1 | 1.15 | 0.31 | 3.3 | – |
| 27 | GPT-6.1 Sol | PI | 5.4 | 0.1–10.7 | 8 | 0 | 0 | 5 | 1 | 0.20 | 0.26 | 2.8 | 11 |
| 28 | Grok 4.7 | Grok Build | 4.8 | 0.4–9.1 | 7 | 0 | 0 | 5 | 0 | 1.33 | 1.54 | 15.6 | 27 |
| 29 | Grok 4.7 | PI | 4.8 | 0.4–9.1 | 7 | 3 | 0 | 5 | 0 | 0.96 | 1.07 | 9.7 | 24 |
| 30 | Claude Sonnet 5 | PI | 4.1 | 0.0–8.7 | 6 | 1 | 0 | 4 | 1 | 1.05 | 1.64 | 9.7 | 34.5 |
| 31 | Gemini 4 Argon | PI | 4.1 | 0.0–9.1 | 6 | 2 | 0 | 3 | 1 | 1.07 | 2.23 | 14.6 | 35 |
| 32 | Nemotron 3 Ultra | PI | 4.1 | 0.0–9.1 | 6 | 1 | 0 | 3 | 1 | 0.11 | 0.57 | 1.7 | 19 |
| 33 | Nemotron 3.5 Lightning | PI | 4.1 | 0.4–7.8 | 6 | 7 | 0 | 5 | 0 | 0.01 | 1.81 | 2.1 | 34 |
| 34 | Claude Sonnet 5.5 | PI | 3.4 | 0.5–6.3 | 5 | 1 | 0 | 5 | 0 | 2.20 | 3.94 | 17.8 | 39 |
| 35 | GLM 5.3 | PI | 3.4 | 0.0–7.9 | 5 | 3 | 0 | 3 | 1 | 0.21 | 0.40 | 3.7 | 23 |
| 36 | Mistral Large 4 | PI | 1.4 | 0.0–3.3 | 2 | 1 | 0 | 2 | 0 | 0.14 | 0.61 | 9.9 | 19 |
| 37 | DeepSeek V4.1 Flash | PI | 0.0 | 0.0–0.0 | 0 | 22 | 0 | 0 | 0 | 0.26 | 11.25 | 46.4 | 90 |
| 38 | Kimi K3.1 Pro | PI | 0.0 | 0.0–0.0 | 0 | 4 | 0 | 0 | 0 | 0.15 | 0.18 | 1.2 | 10 |