

The Generalist Strikes Back:\\A General LLM Agent Outperforms Specialist Bioinformatics Agents on Free-Form, End-to-End Benchmark Analysis

Xiaoyu Zhang

xiaoyu@csusm.edu

California State University, San Marcos <https://orcid.org/0000-0001-8332-2547>

Research Article

Keywords: Large language Model, Bioinformatics Agents, Bioinformatics Benchmark, LLM Agents

Posted Date: October 28th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11485353/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

The Generalist Strikes Back: A General LLM Agent Outperforms Specialist Bioinformatics Agents on Free-Form, End-to-End Benchmark Analysis

Xiaoyu Zhang

California State University San Marcos, San Marcos, CA, USA

xiaoyu@csusm.edu

Abstract

Specialist agents such as Biomni, STELLA and ToolsGenie wrap large language models in curated tools and multi-agent scaffolds for bioinformatics. We asked whether a current frontier model in an off-the-shelf coding harness still needs them. Claude Opus 5.5 (high effort) in Claude Code solved all 244 tasks of PromptBio-Bench, whose free-form output files are scored against expert references. It was run with and without 667 bioinformatics Agent Skills (bioSkills, ClawBio). Under the original evaluation framework with GPT-5.6 judges, Claude with skills completed every task and reached 0.869 accuracy, against 0.758 for Biomni and ToolsGenie and 0.717 for STELLA (paired McNemar $p \leq 2 \times 10^{-4}$). Its failures exposed eleven evaluator defects and several incorrect reference answers. With these defects fixed, Claude reached 0.918 to 0.926 with skills and 0.873 to 0.902 without, across GPT and Claude judges that agreed on 95 to 98% of tasks. Skills added only 2.5 to 4.5 points, yet transcripts showed the agent never invoked one; the gap traces to run-to-run variation in a few judgement calls. Of the 18 tasks still failed, 11 trace to defective references or unstated settings, and most of the rest are defensible alternatives to the reference. A general-purpose frontier model now matches or exceeds purpose-built bioinformatics agents on realistic analyses; benchmark reliability, not agent capability, has become the main limit on measuring progress.

1 Introduction

LLM-based agents that plan analyses, write and execute code, and iterate on their own results are reshaping computational biology [1–3]. A first generation of systems demonstrated automated omics workflows [4, 5]. A second generation of general-purpose biomedical agents expanded the action space and the scaffolding around the model: Biomni integrates a curated biomedical tool and database space with ReAct-style execution [6]; STELLA emphasises self-evolving reasoning strategies and tool creation [7]; ToolsGenie, part of the PromptBio platform, prioritises robust execution through managed Docker environments and a modular multi-agent design [8]; and BioMedAgent and CellVoyager target tool-aware or domain-specific discovery [9, 10].

Benchmarks have followed. LAB-Bench probes biological knowledge and research reasoning [11], BixBench introduced expert-curated analytical questions over real datasets [12], and BioML-bench evaluates end-to-end biomedical machine learning [13]. Several of these benchmarks resolve answers to short, closed forms. On BixBench-Verified-50, a curated subset of BixBench graded with multiple-choice answers, skill-augmented frontier agents reach up to 98% accuracy, nearly saturating

the benchmark [14]. PromptBio-Bench [15] is deliberately harder to saturate: its 244 tasks span bioinformatics and data science at three difficulty levels. Each task requires the agent to produce *output files*, such as alignments, variant calls, count matrices, tables, figures and text, whose format and content are free. These files are scored against expert reference files by a modular evaluator that combines format-specific comparison handlers, quantitative similarity metrics and an LLM judge.

The PromptBio-Bench authors evaluated Biomni, STELLA and ToolsGenie, all backed by Claude Sonnet 4.6, and found accuracies of 0.72 to 0.76 that declined with difficulty [15]. They noted as a limitation that native LLM coding harnesses, such as Claude Code or Codex, were not evaluated. Comparing those harnesses with domain-oriented frameworks is needed to clarify what bioinformatics-specific scaffolding adds. Meanwhile, model providers have introduced *Agent Skills*: packaged, on-demand instructions and code that a general agent can load when relevant [16]. Community libraries such as bioSkills [17] and ClawBio [18] now encode hundreds of bioinformatics procedures in this form, raising a second question: does a strong general model still benefit from domain skills?

This study addresses both questions on PromptBio-Bench. Our contributions are:

1. An evaluation of a frontier general-purpose model, Claude Opus 5.5 at high effort in Claude Code [19, 20], with and without 667 bioinformatics skills, under the same protocol as the original benchmark (one attempt per task, 1-hour limit).
2. A head-to-head comparison with Biomni, STELLA and ToolsGenie under the original PromptBio-Bench evaluation framework, using more recent judge models.
3. A failure analysis that identified eleven evaluator defects and several reference-answer defects, followed by a corrected evaluator and a re-score with two independent judge families (GPT-5.6-sol/6.1-sol and Claude Opus 5.5), including an analysis of judge agreement.
4. A controlled comparison of the skills and no-skills configurations, together with transcript evidence on whether skills were actually used.
5. An analysis of the tasks that still fail, assessing how good the agent’s answers were even when they did not match the reference.

2 Methods

2.1 Benchmark

PromptBio-Bench [15] contains 244 task capsules: 131 bioinformatics tasks (track A) and 113 data-science tasks (track B), with 60 low-, 139 medium- and 45 high-difficulty tasks. Each capsule provides a natural-language question, input files, the expected output file names and formats, an expert reference answer, and a per-file evaluation guideline. Task descriptions specify the analytical goal but usually not the tools or parameters. We used the public task capsules and evaluation code without modifying the tasks or references.

2.2 Agent configuration

Each task was solved by an independent Claude Code subagent running Claude Opus 5.5 at high effort. Both runs started from the same top-level instruction, given to an orchestrating Claude Code session. That instruction asked the session to create one subagent per task, run up to eight in parallel, keep a resumable log, and require each subagent to plan the analysis, solve it as rigorously

as possible in the provided conda environment, use the web for up-to-date tools, keep scripts and notes in a `tmp` folder and the expected files in an `output` folder, and stop at the task’s 3,600-s timeout. Each orchestrator then wrote its own detailed per-task prompt and a workflow script that launched the subagents on a Linux workstation. The two orchestrators covered all of these requirements but worded the per-task prompt differently, and each added guidance of its own (Supplementary Note S1). The skills-run prompt asks for “default/best-practice parameters unless the question says otherwise” and a sanity check of the results; the no-skills prompt asks the agent to inspect the input data first and to “pick sensible defaults” where the question leaves choices open. Both prompts are released with the agent outputs. Each subagent worked in a shared conda environment, into which it could install missing packages (serialised by a lock), and downloaded large public references to a shared cache. Only the final expected files were placed in the output folder, and a 3,600-s wall-clock limit applied to each task. Agents were forbidden to read other task folders, reference answers, reference scripts or evaluation guidelines. As in the original study, one attempt was made per task. Attempts interrupted by subscription usage limits were discarded and restarted from scratch (seven tasks in the skills run), and the time budget was cumulative across attempts in the no-skills run.

Two configurations were run on the same 244 tasks:

- **Claude with skills.** 667 Agent Skills were installed in the project’s skill directory: the 561 skills of bioSkills (63 categories, from sequence I/O to single-cell, spatial, proteomics and clinical workflows) [17] and 106 skills from ClawBio, including its orchestrator [18]. Skill names and descriptions are exposed to the model, which may load any skill on demand.
- **Claude without skills.** The same harness, top-level instruction, model and environment, with no skills installed; its per-task prompt was written by its own orchestrator, as described above.

The specialist agents’ results were taken from the PromptBio-Bench release. Biomni v0.0.8 and ToolsGenie v3.2.0 used Claude Sonnet 4.6. STELLA v1.0.0 combined Claude Sonnet 4.6 with Gemini 2.5 Pro. All were run in Docker with a 1-hour limit [15].

2.3 Evaluation frameworks

Original framework. The PromptBio-Bench evaluator validates each output file, detects its format and asks an LLM to select a comparison strategy (exact, approximate, summary, semantic, numeric, functional) and its parameters. A format-specific handler then computes a similarity in $[0, 1]$. For tables, an LLM plans column mapping and row keys, and per-column metrics are averaged; images and free text are scored by an LLM judge. A task’s score is the mean of its file scores, failed or missing outputs score 0, and a task is counted as accurate if its score is at least 0.5. Using this threshold, the per-difficulty accuracies of the three specialist agents reproduce the published values exactly. The original study used GPT-5.4 as the evaluator’s LLM [15]. We scored Claude with skills with the unmodified framework, using the more recent GPT-5.6 through the OpenAI API and, after API credits ran out, GPT-5.6-sol through the Codex CLI for about 145 tasks [21, 22]. During that run we found that one code path ignored the configured model and silently fell back to GPT-5.4; it was routed to the configured model before the final scores were taken.

Fixed framework. A review of every task scoring below 0.5 (Section 3.2) identified evaluator defects that gave correct answers low scores. We corrected them without changing any task, reference or guideline (Table 2). Every fix is general: none is keyed to a task, and all are applied to every

agent and file. Before re-scoring, we replayed every logged comparison that does not need an LLM with the fixed code. Apart from the tasks the fixes target, scores were unchanged, except for the FASTA approximate strategy, which varies by ≤ 0.002 between runs because of hash-order-dependent greedy pairing. One scoring rule involves a deliberate choice. For single-value numeric outputs, the guideline’s multiplicative tolerance (e.g. “within two orders of magnitude”) is applied as a relative-error tolerance of 99, which accepts any positive value up to $100\times$ the reference. We keep the original rule that unmatched reference rows do not penalise numeric table columns.

Judges. With the fixed framework, both Claude configurations were scored twice. The first judge was a GPT model through the Codex CLI: GPT-5.6-sol for 113 (with skills) and 127 (without skills) tasks, and GPT-6.1-sol for the rest, after a switch made for cost. The second judge was Claude Opus 5.5 through the Claude Code CLI in a tool-less, settings-free headless mode, with images passed as image blocks and structured output enforced by a JSON schema. To limit self-preference, the Claude judge only selects strategies and plans comparisons for structured files and judges images and text; the similarity of structured outputs is computed deterministically by the handlers.

2.4 Statistics

Paired comparisons between configurations on the same tasks use exact McNemar tests on pass/fail outcomes [23] and Wilcoxon signed-rank tests on task similarity. Confidence intervals for accuracy differences come from 10,000 bootstrap resamples of tasks. Agreement between judges is reported as raw pass/fail agreement, Cohen’s κ [24] and Pearson’s r on task scores.

2.5 Failure analysis

For each task below 0.5 we compared the reference answer, the reference script, the agent output, the agent’s notes and scripts, and the evaluator log. Where needed, we re-ran the reference script on the current inputs in an isolated copy. Each failure was assigned one primary class: *scorer defect*, where the answer is correct or within tolerance but scored low; *reference defect*, where the reference is wrong or outdated; *unstated setting*, where the reference depends on a parameter or preprocessing choice the question does not give; or *agent choice*, where the agent’s own decision caused the mismatch.

3 Results

3.1 Under the original evaluation framework, Claude with skills outperforms all three specialist agents

Claude with skills produced a valid output for all 244 tasks (completion 1.00, vs. 0.99, 0.98 and 0.88 for Biomni, ToolsGenie and STELLA). Its accuracy was 0.869, against 0.758 for Biomni and ToolsGenie and 0.717 for STELLA (Table 1, Figure 1). Its mean similarity was 0.793, against 0.648 to 0.717. The advantage held at every difficulty level and was largest where the specialist agents plateau: on medium tasks Claude reached 0.878, while all three specialists stayed at 0.72 to 0.73, and on high-difficulty tasks it reached 0.756, against 0.56 to 0.69. It was also concentrated in the bioinformatics track: track A accuracy was 0.817, against 0.618 to 0.641, whereas on track B, where Biomni already reaches 0.920, Claude reached 0.929.

On paired tasks, Claude with skills passed 39 tasks that Biomni failed while failing 12 that Biomni passed (exact McNemar $p = 2.0 \times 10^{-4}$). The corresponding counts were 38 vs. 11 against

Table 1: Performance under the original PromptBio-Bench evaluation framework. Specialist results are from the PromptBio-Bench release (GPT-5.4 judge); Claude with skills was scored with the same framework using GPT-5.6 and GPT-5.6-sol judges. Accuracy is the share of tasks with similarity ≥ 0.5 ; failed tasks score 0. Backbones: Biomni and ToolsGenie, Claude Sonnet 4.6; STELLA, Claude Sonnet 4.6 with Gemini 2.5 Pro. Time is wall-clock seconds per task.

Agent	Completion	Accuracy				Mean sim.	Time (s)
		Low	Medium	High	All		
Biomni	0.992	0.900	0.727	0.667	0.758	0.699	171 / 82
STELLA	0.881	0.800	0.734	0.556	0.717	0.648	275 / 74
ToolsGenie	0.980	0.900	0.719	0.689	0.758	0.717	218 / 100
Claude Opus 5.5 + skills	1.000	0.933	0.878	0.756	0.869	0.793	171 / 39

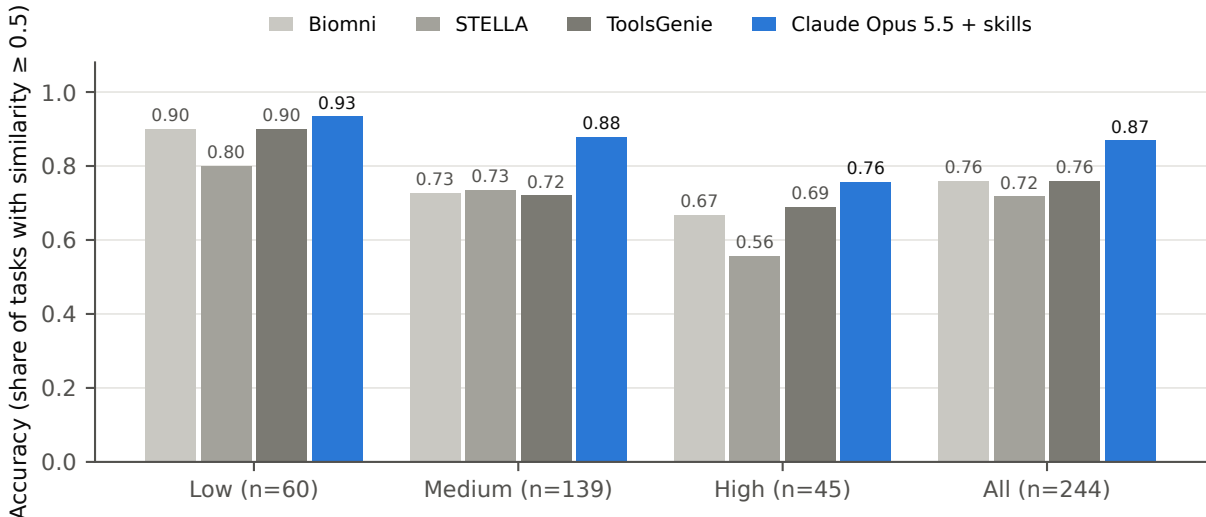


Figure 1: Accuracy by task difficulty under the original PromptBio-Bench evaluation framework. Specialist agents in grey, Claude Opus 5.5 with bioinformatics skills in blue.

ToolsGenie ($p = 1.4 \times 10^{-4}$) and 50 vs. 13 against STELLA ($p = 3.0 \times 10^{-6}$). Twenty-seven tasks were failed by all three specialists, and Claude passed twelve of them. Taking the best of the three specialists task by task, an oracle that no real system can achieve, reaches 0.889, only two points above Claude alone. Claude’s mean runtime per task was 171 s (median 39 s), the same mean as Biomni (171 s; median 82 s) and below ToolsGenie (218 s) and STELLA (275 s; Supplementary Table S4).

3.2 Failure analysis reveals scorer and reference defects

Thirty-two tasks scored below 0.5 for Claude with skills under the original framework. Inspecting each one changed how they should be read (Supplementary Table S2). Only 8 of the 32 (3.3% of all tasks) were mistakes or poor choices by the agent. Sixteen were *scorer defects*, in which the agent’s answer was correct or within the guideline’s tolerance. Three were *reference defects*, in which the reference answer itself was wrong. Five depended on *settings the question never states*. Several scorer defects were absolute: in a-14-2, a-14-3 and a-14-5 the agent’s residue and motif sets were identical to the reference, and in a-3-5 all three output files were byte-identical to it, yet these tasks

Table 2: Evaluator defects found by the failure analysis and their fixes. All fixes are task-independent and applied to every agent; examples are the claude-skills tasks they affected.

Defect	Fix	Example tasks
All columns are row keys, or all value columns ignored or unscorable → score 0	Score by F1 of matched row keys, or of whole rows if not keyed	a-14-2/3/5, b-10-9, a-3-6
Empty reference and empty candidate → “invalid”, 0	Score 1.0 when both contain no data records	a-3-5, a-7-8
<code>astype(bool)</code> turns “No” into <code>True</code>	Explicit yes/no/true/false mapping	b-4-15
<code>summary</code> strategy on small result tables	Value-level comparison for tables ≤ 100 cells	b-7-4, b-7-5, b-11-4
FASTQ records matched only by read name	Fall back to identical sequence and quality	a-1-14
FASTA <code>by_name</code> with no shared IDs	Fall back to sequence matching	a-6-3
Whitespace normalisation deleted newlines	Keep line structure	a-4-4
Numeric tolerance checked against clipped similarity	Use raw relative error	a-7-9
<code>pearsonr(nan_policy=...)</code> unsupported by installed SciPy → 0	Apply NaN policy before <code>pearsonr</code>	a-6-4, b-6-7
Format detection ignored the configured judge	Inherit the evaluator’s model	a-12-1, a-3-5
Composite keys, labels and float keys failed on formatting	Normalised keys; one-sided affixes; precision-matched floats; <code>partition</code> and <code>compare_abs</code> column options	b-2-13, a-2-8, a-5-10, a-8-1

scored 0 or 0.33. The same defects affected every agent: 15 of the 244 tasks were failed by all four systems, and 13 of these were scorer defects or unstated settings.

Table 2 lists the evaluator defects we fixed. Most concern table handling: tables whose only columns are row identifiers left nothing to compare; small result tables were scored by distributional summaries instead of by value; and a “No” string became `True` under `astype(bool)`. Others concern sequence files matched by identifier rather than content, numeric checks against a clipped similarity, and two silent infrastructure faults. One was a SciPy API change that set every Pearson-scored column to 0; the other was format detection that ignored the configured judge and called a credit-less API. Re-running the reference scripts on the current inputs exposed reference defects. In a-9-9 the reference drops 89 to 95% of reads by coercing “;”-joined counts to NaN. In a-10-2 an allele-coding mismatch leaves only 514 of 2,000 SNPs scored and produces an AUC of 0.38, worse than chance. In b-3-15 the reference image was generated from different data. These findings motivated the corrected evaluator and the use of more recent judges.

3.3 Re-scoring with the fixed evaluator: skills help little, and are not used

Table 3 and Figure 2 summarise the re-scored results. With the fixed evaluator, Claude with skills reached 0.918 with the GPT judge and 0.926 with the Claude judge. Compared with the original framework, 18 tasks moved from fail to pass and 4 from pass to fail under the Claude judge. Only one of the four new failures, b-9-4, is a genuine agent error that the old `summary` strategy

had passed on table shape alone. The other three (a-9-6, b-3-8, b-8-7) are reference defects that value-level comparison now exposes (Section 3.4). Claude without skills reached 0.873 and 0.902. Both configurations exceed every specialist agent’s 0.717 to 0.758, although the specialists were scored only with the original framework (see Limitations).

Judge agreement. The two judge families agreed on pass/fail (Supplementary Table S3) for 97.5% of tasks with skills ($\kappa = 0.83$, $r = 0.95$) and 94.7% without ($\kappa = 0.73$, $r = 0.91$; Figure 3). The disagreements are informative. Four Claude-skills tasks pass only under the Claude judge, mostly because of evaluator fixes made just before that run: guidance on mirrored heatmaps, comment-line parsing and key normalisation. Two tasks fail only under the Claude judge, both image tasks within 0.1 of the threshold whose underlying cause is a reference defect. The near-identical aggregate accuracy of two independent judge families after the fixes suggests that the remaining scores reflect output quality rather than judge idiosyncrasy.

With vs. without skills. Skills improved accuracy by 4.5 points with the GPT judge (95% CI +2.0 to +7.4; 11 wins, 0 losses; McNemar $p < 0.001$; Wilcoxon $p = 0.05$). With the Claude judge the gain was 2.5 points (95% CI +0.4 to +4.9; 7 wins, 1 loss; McNemar $p = 0.07$; Wilcoxon $p = 0.35$). Gains appeared on medium- and high-difficulty tasks; on low-difficulty tasks both configurations scored 0.983. The most striking finding, however, comes from the agent transcripts. The skill catalogue (names and descriptions) was present in the context of 441 of the 444 subagent sessions of the skills run, yet *no session invoked a skill or read a skill file*. No skill content was therefore ever used, and no agent even mentioned a skill in its visible output. The catalogue still had a cost: the skills run consumed 101.0 M tokens against 74.6 M without skills (+35%, almost all cached prompt tokens) and took longer (median 39 vs. 31 s per task).

Where the two configurations differ. To explain the gap without skill use, we examined every task that only one configuration passed under either judge (13 tasks). Seven are scoring variance rather than different answers: the two agents’ outputs scored alike, or both passed, under one judge, and differed only under the other judge or within 0.05 of the threshold (a-2-8, a-5-6, b-10-9, b-8-9, b-4-12, b-4-13, a-15-7). The other six are consistent under both judges, and in each the two agents ran essentially the same analysis but made a different judgement call (Table 4). In five of the six, the no-skills agent chose a more elaborate, non-default option: adduct-aware mass matching, spot-level quality filtering, tuned mass tolerances, a published tie-breaking rule, or a tuned decision tree. These choices are scientifically reasonable but moved it away from the reference. In b-7-7 the skills agent explicitly kept the default model “per ‘default parameters unless stated’”, quoting its own prompt. In a-9-1 the pattern reversed, and the skills agent’s extra call-quality filter brought it closer to the reference. The gap is therefore best explained by run-to-run variation in such judgement calls, plausibly nudged by the orchestrator-written prompts’ different emphasis on defaults, rather than by the skills. Our single-run design cannot apportion it between these sources.

3.4 The tasks that still fail: matched less often, but usually reasoned well

Under the best scorer, the fixed evaluator with the Claude judge, Claude with skills failed 18 of 244 tasks (7.4%; Table 5). Claude without skills failed 17 of the same 18. Only seven of the 18 failures trace to the agent’s own choices (2.9% of all tasks). Seven are reference defects, and four depend on settings the task never states.

Table 3: Accuracy (mean similarity) of Claude Opus 5.5 with and without bioinformatics skills under the original and the fixed evaluation frameworks. “Wins/losses” counts tasks passed by one configuration only.

Evaluator	Judge	Config.	Low	Medium	High	All	Wins/losses
Original	GPT-5.6 / 5.6-sol	skills	0.933	0.878	0.756	0.869 (0.793)	
Fixed	GPT-5.6-sol / 6.1-sol	skills	0.983	0.914	0.844	0.918 (0.863)	11 / 0
		no skills	0.983	0.849	0.800	0.873 (0.838)	
Fixed	Claude Opus 5.5	skills	0.983	0.921	0.867	0.926 (0.866)	7 / 1
		no skills	0.983	0.892	0.822	0.902 (0.849)	

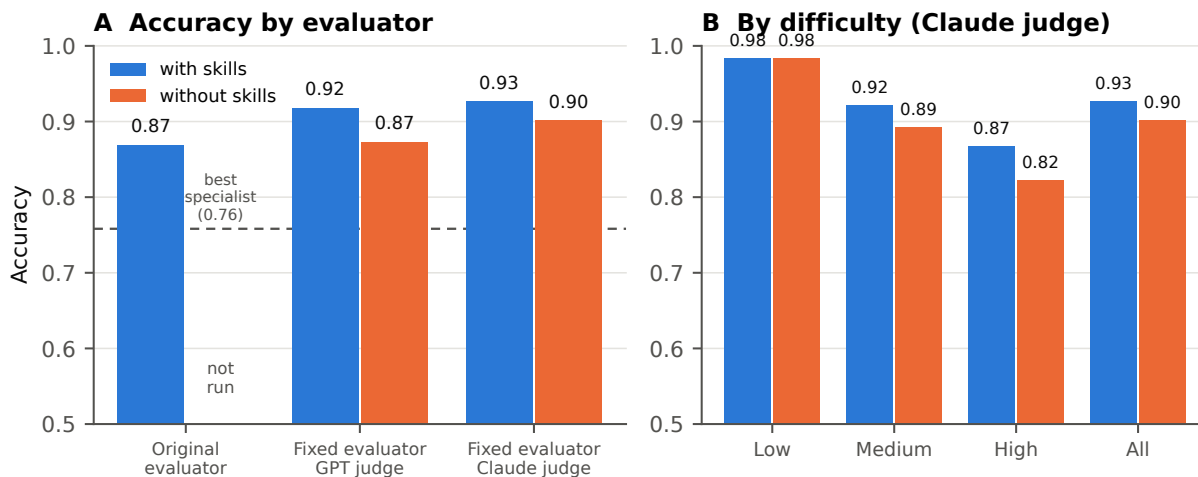


Figure 2: (A) Accuracy of Claude Opus 5.5 with and without skills under the original evaluator and the fixed evaluator with two judge families; the dashed line marks the best specialist agent under the original evaluator. The no-skills configuration was not scored with the original evaluator. (B) Accuracy by difficulty under the fixed evaluator with the Claude Opus 5.5 judge.

Reference defects (7). In each of these, the agent’s answer is correct or more defensible than the reference.

- **Data the reference mishandled.** The agent’s Morisita-Horn index for a-9-9 (0.882) matches vegan, whereas the reference discards most reads [25]. In a-10-2 the agent detected the A/B vs. ACGT allele mismatch, reproduced the reference’s AUC of 0.381, rejected it as worse than chance, and harmonised the alleles to obtain 0.750.
- **A reference that answers a different question.** In b-8-7 the reference predicts a column named **Target** while keeping the near-duplicate **LengthOfStay** ($r = 0.977$) as a feature, whereas the question asks for length of stay. The reference pipeline, run on **LengthOfStay**, reproduces the agent’s MAEs to within 1%.
- **Outdated reference images.** The reference PNGs for b-3-15 and b-3-8 were generated from different data. The reference scripts run on the current inputs select exactly the agent’s genes; in b-3-8 the reference annotates a “top-5” gene whose adjusted p is 0.95.
- **A different unit of counting.** In a-9-6 the reference counts contig rows (about 3.5 per

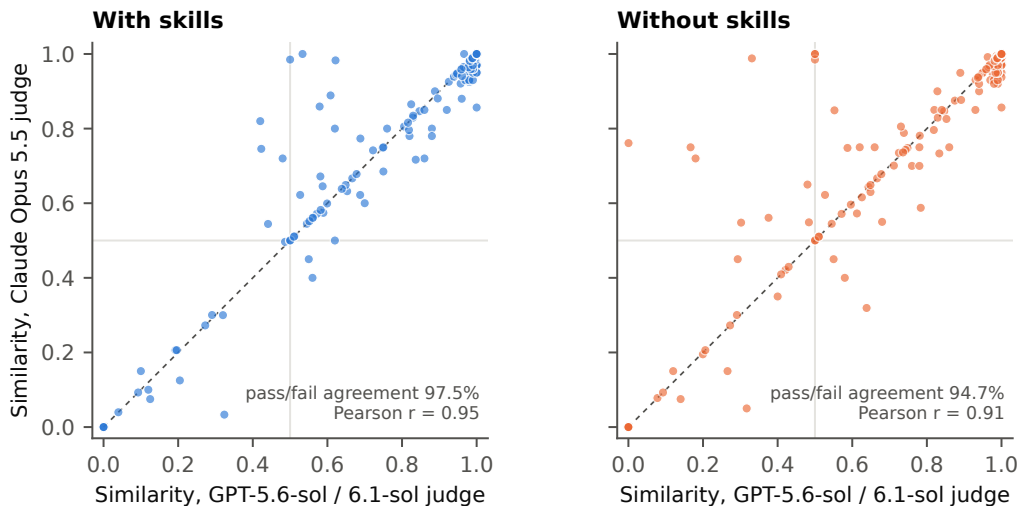


Figure 3: Task-level similarity under the two judges with the fixed evaluator. Grey lines mark the 0.5 pass threshold; the dashed diagonal marks equality.

Table 4: Tasks whose pass/fail differs between Claude with and without skills under both judges. Scores are under the fixed evaluator with the Claude Opus 5.5 judge. Both agents used the same overall method; the difference is a single judgement call.

Task	Skills	No skills	Judgement call (skills vs. no skills)
a-8-1	1.00	0.00	Query m/z matched directly to neutral mass, as the task states, vs. assuming $[M+H]^+$ adducts
a-7-8	1.00	0.00	DIA-NN defaults (no precursor at 1% FDR, like the reference) vs. tuned 10/5 ppm mass accuracy (3 precursors)
b-6-4	1.00	0.00	$k = 2$ and $k = 3$ tied (cophenetic 1.0): first maximum ($k = 2$, as the reference) vs. Brunet’s rule ($k = 3$)
a-5-4	0.95	0.42	All 1,310 spots vs. spot QC filtering (1,172 kept), which shifts every Moran’s I
b-7-7	0.82	0.43	Default unpruned tree (accuracy 0.671; reference 0.65) vs. CV-tuned depth-3 tree (0.742) with different importances
a-9-1	0.58	0.41	Call-quality filter applied (fewer peptides, closer to the reference) vs. all calls kept

cell) as cells, whereas the agent counted cells. The zero/non-zero pattern of the heatmap is identical.

- **A methodologically questionable reference.** In a-7-7 the reference reports all 375 rank-1 peptide-spectrum matches, including 164 decoys, without FDR control. The agent applied the standard 1% target-decoy FDR [26, 27].

Unstated settings (4). The references for a-3-1, a-11-6, a-15-9 and a-4-6 depend on choices absent from the question: log-transformation and per-sample standardisation before ComBat [28], a whole-genome enrichment universe with term simplification [29], non-default WGCNA module parameters on near-noise data [30], and a non-default omicwas test. In each, the agent made a standard, defensible choice, and every agent scored low.

Agent choices (7). The agent’s errors were mostly ones of judgement rather than competence.

Table 5: Tasks failed by Claude with skills under the fixed evaluator and the Claude Opus 5.5 judge. Scores are task similarities; “GPT” is the fixed evaluator with the GPT-5.6-sol/6.1-sol judge.

Task	Class	Claude	GPT	No skills	Cause
b-8-4	Agent	0.00	0.00	0.00	ElasticNet on unscaled features; its scaled run matched the reference
a-11-7	Agent	0.03	0.32	0.05	Correlated with raw instead of inverted essentiality (top/bottom swapped)
a-9-4	Agent	0.09	0.09	0.09	xCell <code>rnaseq=FALSE</code> ; default run (100% match) left unsubmitted
a-4-4	Agent	0.15	0.10	0.45	MEME switched from <code>zoops+revcomp</code> to <code>anr</code>
a-6-2	Agent	0.21	0.19	0.20	Hand-built DADA2 instead of nf-core/ampliseq (84 vs. 244 ASVs)
b-9-4	Agent	0.30	0.29	0.30	$\lambda_{\min}$ kept 31 features; λ_{1se} run matched exactly
a-15-7	Agent	0.50*	0.49	0.55	Shortened iClusterPlus λ grid; 39% cluster agreement
a-9-9	Reference	0.00	0.00	0.00	Reference drops 89 to 95% of reads; agent’s value is correct
a-7-7	Reference	0.04	0.04	0.08	Reference includes 164 decoy PSMs, no FDR control
a-10-2	Reference	0.08	0.12	0.08	Allele-coding mismatch: reference AUC 0.38 vs. agent 0.75
b-8-7	Reference	0.21	0.20	0.21	Reference predicts <code>Target</code> using <code>LengthOfStay</code> as a feature
b-3-15	Reference	0.30	0.32	0.35	Stale reference image
b-3-8	Reference	0.40	0.56	0.40	Stale reference image (annotates a gene with adj. $p = 0.95$)
a-9-6	Reference	0.45	0.55	0.45	Reference counts contigs as cells (13 vs. 4)
a-3-1	Unstated	0.00	0.00	0.00	<code>log2</code> , per-sample z-scoring and ComBat not stated
a-11-6	Unstated	0.10	0.12	0.15	Whole-genome universe and <code>simplify()</code> not stated
a-15-9	Unstated	0.12	0.20	0.15	WGCNA parameters not stated; near-noise data
a-4-6	Unstated	0.27	0.27	0.27	<code>nls.logit + regularize</code> not stated

*0.496.

- **A matching variant computed but not submitted (4 tasks).** The agent’s notes show a sensitivity run that reproduces the reference, after which it submitted an alternative it argued was better. These were xCell with `rnaseq=FALSE` for microarray-like data (a-9-4) [31], unstandardised ElasticNet (b-8-4), the CV-optimal $\lambda_{\min}$ rather than λ_{1se} before an unpenalised Cox refit (b-9-4) [32], and an `anr` MEME model (a-4-4) [33].
- **Pipeline shortcuts (2 tasks).** The agent hand-built DADA2 instead of running nf-core/ampliseq (a-6-2) [34, 35], and shortened the iClusterPlus λ grid (a-15-7, which misses the threshold by 0.004) [36].
- **A missed sign convention (1 task).** In a-11-7 the agent correlated expression against raw rather than inverted essentiality, which swaps the top- and bottom-ranked genes.

In short, among Claude’s failures with the best scorer, a strict reading finds roughly one clear analytical error (a-11-7) and a handful of over-confident deviations from defaults. A reviewer reading the agent’s notes would accept most of the remaining failed outputs as competent analyses.

4 Discussion

A generalist now matches or beats specialist scaffolds. On a benchmark designed around realistic, file-producing bioinformatics workflows, a frontier model in a general-purpose coding harness completed every task and was more accurate than three purpose-built agents. This held under the benchmark’s own evaluator before any of our fixes. The advantage was largest on medium- and high-difficulty bioinformatics tasks, where the specialists plateau at about 0.72 and below.

We attribute it mainly to model capability combined with a capable but generic harness. The agent could install and run any tool, fetch references and inspect intermediate results, so domain knowledge came from the model rather than from a curated action space. Runtime was comparable to the fastest specialist. These results do not show that scaffolding is useless, but they do show that the benefit of a domain-specific scaffold must now be weighed against a much stronger baseline.

Are bioinformatics skills necessary? For Claude Opus 5.5 on PromptBio-Bench, the answer is largely no. The skills configuration was 2.5 to 4.5 points more accurate depending on the judge. That gain is statistically clear with one judge and borderline with the other, and the model never loaded a single skill. The measured benefit therefore cannot come from skill content. Task-level inspection traces it instead to a handful of judgement calls, such as adduct handling, quality filtering and tie-breaking, in which the no-skills agent more often chose a non-default option that diverged from the reference. That is run-to-run variation, possibly shaped by the differently worded orchestrator prompts, rather than an effect of skills. Meanwhile, the catalogue increased token consumption by about a third. A controlled test would rerun both configurations from a fixed per-task prompt, with replicates. This contrasts with BixBench-Verified-50, where skills were part of the configurations that nearly saturated the benchmark [14]. The difference may reflect the benchmarks: multiple-choice questions reward recalling a specific procedure, whereas PromptBio-Bench’s free-form outputs reward executing standard tools correctly, which the model does unaided. Skills may matter more for weaker models, proprietary or lab-specific procedures, and tasks outside the model’s training distribution. They may also matter when the harness actively routes to them, which we did not test.

Benchmark reliability is now the bottleneck. Our analysis found more scorer and reference defects than agent errors among the original failures. Fixing eleven evaluator defects moved accuracy by five to six points, more than the difference between using skills or not, and at least seven reference defects remain in the benchmark. The original paper acknowledged that a single reference captures one analytical path among several valid ones [15]. Our results show this concretely: four failures depend on unstated settings, and four agent-choice failures are defensible alternatives that the agent itself had computed and set aside. LLM judges add variance; the two judge families disagreed on 2.5 to 5% of tasks [37, 38]. Free-form benchmarks remain essential precisely because closed-form ones saturate. To track frontier agents, they will need multiple references or rubric-based equivalence, deterministic checks for figures backed by data, routine re-execution of reference scripts, and replicate scoring.

Can a frontier LLM replace Biomni or ToolsGenie? For well-specified analyses of the kind PromptBio-Bench contains, our results suggest that a frontier model in a general harness can replace these systems on accuracy and completion. Specialist platforms still offer things a benchmark does not measure: curated and validated environments, provenance and audit trails, data governance and on-premise deployment, user interfaces for non-programmers, and cost control through smaller backbones. The comparison also confounds architecture with model generation. The specialists ran on Claude Sonnet 4.6, whereas Claude Code ran Claude Opus 5.5. The most informative next experiment is to run Biomni, STELLA and ToolsGenie with the same frontier backbone and re-score all systems with the fixed evaluator. That would show whether their scaffolds add value on top of a frontier model, or whether, as our results suggest, the model has absorbed most of what the scaffold provided.

Limitations.

1. **The specialist agents were not re-scored.** They were scored only with the original framework and an older judge, and their outputs were not available to us. The comparisons with fixed-evaluator scores are therefore favourable to Claude, and only the original-framework comparison is like for like.
2. **Different backbones.** Backbone model and release date differ between Claude and the specialists, as discussed above.
3. **One run per configuration.** Each configuration was run once, so run-to-run variance is unquantified; this matters most for the small skills effect.
4. **Orchestrator-written prompts.** The two runs received the same top-level instruction, but each orchestrating session wrote its own per-task prompt (Supplementary Note S1). The skills comparison is therefore not controlled at the prompt level.
5. **Mixed GPT judge.** The GPT judge was a mix of GPT-5.6-sol and GPT-6.1-sol. The last fixes (delimiter normalisation, one-sided key affixes and the single-value numeric rule) were applied before the Claude-judge run but not the GPT-judge run.
6. **Evaluator choices made by us.** The evaluator fixes and two scoring rules were designed while inspecting one agent’s outputs. Although every fix is general and was validated by replay, they were not pre-registered.
7. **Possible self-preference.** The Claude judge may favour Claude outputs; this is mitigated by deterministic scoring of structured files and by the close agreement with the GPT judge.
8. **Possible contamination.** We cannot exclude that the public PromptBio-Bench materials were in the model’s training data.
9. **AI assistance.** The failure analyses were performed with the assistance of Claude-based agents and verified by spot checks, not by independent human experts.

5 Conclusion

A general-purpose frontier LLM, run in an off-the-shelf coding harness, outperformed three state-of-the-art specialist bioinformatics agents on PromptBio-Bench’s free-form, end-to-end tasks, with and without bioinformatics skills. Skills were never actually invoked, and the small accuracy gap between the configurations traces to run-to-run variation in judgement calls rather than to skill use. Most of the remaining failures reflect benchmark defects or underspecified tasks rather than analytical errors. As frontier models approach the ceiling of current benchmarks, the community’s priority should shift toward more robust references, evaluators and replicate protocols that can still separate good analyses from great ones.

Data and code availability

PromptBio-Bench tasks and the original evaluator are available from the PromptBio-Bench release (<https://github.com/PromptBio/promptbio-bench>; <https://huggingface.co/datasets/promptbio-ai/promptbio-bench-data>). The corrected evaluator, the Claude Code judge backend, the agent outputs, per-task evaluation logs for all scoring runs, and the scripts used to produce the tables and figures will be released with this preprint (<https://github.com/>

xiaoyu12/promptbio-bench). Per-task scores for every agent and evaluator are given in Supplementary Table S5 and as `supplementary_table_S5_scores.csv`.

Acknowledgements

Agent runs and judge evaluations used Claude Code and Codex CLI subscriptions. Claude-based agents assisted with the failure analysis, evaluator debugging and drafting of this manuscript; the author reviewed and takes responsibility for all content.

Competing interests

The author declares no competing interests.

References

- [1] Hanchen Wang et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620:47–60, 2023.
- [2] Juexiao Zhou, Jindong Jiang, Zhongyi Han, Zhe Wang, and Xin Gao. Streamline automated biomedical discoveries with agentic bioinformatics. *Briefings in Bioinformatics*, 26:bbaf505, 2025.
- [3] C. Qi et al. Artificial intelligence agents for biological research: a survey. *Briefings in Bioinformatics*, 27:bbag075, 2026.
- [4] Juexiao Zhou et al. An AI agent for fully automated multi-omic analyses. *Advanced Science*, 11:2407094, 2024.
- [5] Houcheng Su, Weicai Long, and Yanlin Zhang. BioMaster: Multi-agent system for automated bioinformatics analysis workflow. *bioRxiv*, 2025. doi: 10.1101/2025.01.23.634608.
- [6] Kexin Huang et al. Biomni: A general-purpose biomedical AI agent. *bioRxiv*, 2025. doi: 10.1101/2025.05.30.656746.
- [7] Ruofan Jin et al. STELLA: Towards a biomedical world model with self-evolving multimodal agents. *bioRxiv*, 2025. doi: 10.1101/2025.07.01.662467.
- [8] Minzhe Zhang et al. PromptBio: A multi-agent AI platform for bioinformatics data analysis. *bioRxiv*, 2025. doi: 10.1101/2025.07.05.663295.
- [9] Dechao Bu et al. Empowering AI data scientists using a multi-agent LLM framework with self-evolving capabilities for autonomous, tool-aware biomedical data analyses. *Nature Biomedical Engineering*, 2026. doi: 10.1038/s41551-026-01634-6.
- [10] Samuel Alber et al. CellVoyager: AI CompBio agent generates new insights by autonomously analyzing biological data. *Nature Methods*, 23:749–759, 2026.
- [11] Jon M. Laurent et al. LAB-Bench: Measuring capabilities of language models for biology research, 2024.

- [12] Ludovico Mitchener et al. BixBench: a comprehensive benchmark for LLM-based agents in computational biology, 2025.
- [13] Henry E. Miller, Matthew Greenig, Benjamin Tenmann, and Bo Wang. BioML-bench: Evaluation of AI agents for end-to-end biomedical ML. *bioRxiv*, 2025. doi: 10.1101/2025.09.01.673319.
- [14] Xiaoyu Zhang. Skill-augmented frontier agents nearly saturate BixBench-Verified-50. *bioRxiv*, 2026. doi: 10.64898/2026.04.28.721523.
- [15] Wenbin Guo, Minzhe Zhang, Bowei Han, Youjia Ma, Yang Leng, Shishir Hebbar, Xiaoyuan Zhou, Wenhao Gu, Xiao Yang, and Shashi Dhar. PromptBio-Bench: Benchmarking LLM-based bioinformatics agents for end-to-end data analysis. *bioRxiv*, 2026. doi: 10.64898/2026.05.05.723092. Version 2.
- [16] Anthropic. Equipping agents for the real world with agent skills. <https://www.anthropic.com/engineering/equipping-agents-for-the-real-world-with-agent-skills>, 2025.
- [17] GPTomics. bioSkills: bioinformatics skills for AI coding agents. <https://github.com/GPTomics/bioSkills>, 2026. MIT License; 561 skills in 63 categories.
- [18] ClawBio contributors. ClawBio: a bioinformatics-native AI agent skill library. <https://github.com/ClawBio/ClawBio>, 2026. MIT License.
- [19] Anthropic. Claude opus 5.5. <https://www.anthropic.com/claude/opus>, 2026.
- [20] Anthropic. Claude code documentation. <https://code.claude.com/docs>, 2026.
- [21] OpenAI. OpenAI models documentation (GPT-5.6, GPT-6.1). <https://platform.openai.com/docs/models>, 2026.
- [22] OpenAI. Codex CLI. <https://github.com/openai/codex>, 2026.
- [23] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12:153–157, 1947.
- [24] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960.
- [25] Henry S. Horn. Measurement of “overlap” in comparative ecological studies. *The American Naturalist*, 100:419–424, 1966.
- [26] Joshua E. Elias and Steven P. Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature Methods*, 4:207–214, 2007.
- [27] Jimmy K. Eng, Tahmina A. Jahan, and Michael R. Hoopmann. Comet: an open-source MS/MS sequence database search tool. *Proteomics*, 13:22–24, 2013.
- [28] W. Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8(1):118–127, 2007.
- [29] Tianzhi Wu et al. clusterProfiler 4.0: A universal enrichment tool for interpreting omics data. *The Innovation*, 2:100141, 2021.
- [30] Peter Langfelder and Steve Horvath. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics*, 9:559, 2008.

- [31] Dvir Aran, Zicheng Hu, and Atul J. Butte. xCell: digitally portraying the tissue cellular heterogeneity landscape. *Genome Biology*, 18:220, 2017.
- [32] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.
- [33] Timothy L. Bailey et al. MEME SUITE: tools for motif discovery and searching. *Nucleic Acids Research*, 37:W202–W208, 2009.
- [34] Benjamin J. Callahan, Paul J. McMurdie, Michael J. Rosen, Andrew W. Han, Amy Jo A. Johnson, and Susan P. Holmes. DADA2: High-resolution sample inference from Illumina amplicon data. *Nature Methods*, 13:581–583, 2016.
- [35] Philip A. Ewels et al. The nf-core framework for community-curated bioinformatics pipelines. *Nature Biotechnology*, 38:276–278, 2020.
- [36] Qianxing Mo et al. Pattern discovery and cancer gene identification in integrated cancer genomic data. *Proceedings of the National Academy of Sciences USA*, 110:4245–4250, 2013.
- [37] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.
- [38] Jiawei Gu et al. A survey on LLM-as-a-judge, 2024.

Supplementary Materials

Supplementary Note S1: agent instructions

Both runs were launched by giving an orchestrating Claude Code session the same top-level instruction, reproduced here verbatim:

The tasks folder contains the 244 benchmark tasks for PromptBio-Bench (<https://doi.org/10.64898/2026.05.05.723092>), a benchmark for evaluating LLM agents on real-world computational biology and bioinformatics workflows.

Each task is independent and its corresponding directory contains a task.json file describing the question, optional input_files and expected_output. For each task, create a subagent to complete the task. Allow multiple subagents to run in parallel (up to 8) if resources allow. Maintain a log of completed tasks so that this session can be resumed for remaining unfinished tasks.

Requirements for the subagent of each task:

- Understand task in the task.json, form a plan, then write and execute any code needed to answer the question as rigorously as possible.
- Use the provided conda environment “bench” for code development and execution
- Use the input data and files you create during analysis.
- Use the web to find up to date tools and packages.
- Create a “tmp” directory in each task folder to save any scripts, intermediate outputs, or notes.
- Create an “output” directory in each task folder to save the expected output file as specified by the “expected_output” in the task.json file.
- timeout_seconds = 3600 for each task. If a task doesn’t complete within the timeout_seconds. Quit the task and mark it as failed.

Each orchestrating session turned this instruction into its own per-task prompt and a workflow script that launched one subagent per task. Table S1 summarises how the two prompts differ. The full prompts are saved as `agent_prompt_template.txt` in each run’s folder, with the workflow scripts (`bench_workflow.js`). Rendered for task b-7-7, the no-skills template matches the prompt recorded in that agent’s transcript word for word.

Supplementary Note S2: verifying skill use

Claude Code records every tool call of every subagent in a JSONL transcript. A skill’s content can reach an agent only through a tool call: the `Skill` tool, or a read of a file under `.claude/skills/`. We scanned all 444 subagent transcripts of the skills run, covering all 244 tasks, for both, and for any mention of a skill in the agent’s visible text or tool inputs. The skill catalogue appeared in the context of 441 sessions. We found no `Skill` call, no access to a skill file and no mention of a skill. As a positive control, the same scan detects every skill call in other sessions, including one call to a built-in charting skill in the no-skills run. The script is released with the outputs (`verify_skill_use.py`).

Table S1: Differences between the per-task prompts written by the two orchestrating sessions.

Element	Claude with skills	Claude without skills
Method guidance	“Use standard, well-established tools and default/best-practice parameters unless the question says otherwise”	“using standard, well-accepted methods and tools . . . Where the question leaves choices open, pick sensible defaults and record them”
Before analysing		“Inspect the input data before deciding on a method”
After analysing	“Sanity-check results before finishing”	–
Written answers		An answer.md file if the question asks for a written answer not captured by a file (98 tasks; never matched to a reference file by the evaluator)
Shared references	Large downloads go to a shared cache, under a per-resource lock	Not mentioned
Integrity	Do not read reference answers, reference scripts, evaluation files or other task folders	The same, plus other agents’ published results online
Common to both	Plan file, bench environment, web for tools, tmp/output folders, exact output names and formats, 6 threads and 7 GB, 3,600-s limit, completion logging	

Table S2: The 32 tasks that Claude with skills failed under the original evaluator, with the class assigned by the failure analysis and the task score before and after the evaluator fixes. Classes: *Scorer*, the answer is correct or within tolerance but was scored low; *Reference*, the reference answer is wrong or outdated; *Unstated*, the reference depends on a setting the question does not give; *Agent*, the agent’s own choice caused the mismatch.

Task	Class	Original	Fixed (GPT)	Fixed (Claude)	Cause
a-1-14	Scorer	0.00	1.00	1.00	FASTQ records matched by read name; sequences and qualities identical
a-3-6	Scorer	0.00	0.42	0.75	316/346 junctions recovered; header and comment lines broke row matching
a-4-4	Scorer	0.00	0.10	0.15	Whitespace option deleted newlines (still fails on MEME settings)
a-6-3	Scorer	0.00	1.00	1.00	Same ORF protein; FASTA matched by ID with different ID styles
a-7-7	Scorer	0.00	0.04	0.04	Only value column ignored; reference also keeps decoys (re-classed as reference)
a-7-8	Scorer	0.00	1.00	1.00	Reference and agent both empty; empty file marked invalid
a-14-2	Scorer	0.00	1.00	1.00	30/30 residues identical; all columns are row keys
a-14-3	Scorer	0.00	1.00	1.00	112/112 residues identical; same defect
a-14-5	Scorer	0.00	1.00	1.00	2/2 motifs identical; same defect
b-7-4	Scorer	0.24	0.99	0.99	AUPRC within ± 0.05 ; summary strategy scored shape and labels
b-10-9	Scorer	0.25	1.00	1.00	Same 30 features (key-only column scored 0); AIC within tolerance
a-3-5	Scorer	0.33	1.00	1.00	All 3 files byte-identical; header-only files scored 0
b-7-5	Scorer	0.46	0.98	0.98	Accuracy within ± 0.05 ; summary penalised extra columns
b-2-13	Scorer	0.48	0.96	0.92	All 4,494 reference pairs present; rows paired by position
b-11-4	Scorer	0.49	1.00	1.00	Estimates within 1%; summary penalised an extra row
b-4-15	Scorer	0.50*	1.00	1.00	“No” cast to True by astype(bool)
a-9-9	Reference	0.00	0.00	0.00	Reference drops 89 to 95% of reads
a-10-2	Reference	0.14	0.12	0.08	Allele-coding mismatch; reference AUC 0.38
b-3-15	Reference	0.46	0.32	0.30	Stale reference image
a-3-1	Unstated	0.00	0.00	0.00	log2, per-sample z-scoring and ComBat not stated
a-11-6	Unstated	0.12	0.12	0.10	Whole-genome universe and simplify() not stated
a-4-6	Unstated	0.27	0.27	0.27	nls.logit + regularize not stated
a-15-9	Unstated	0.36	0.20	0.12	WGCNA parameters not stated; near-noise data
a-11-2	Unstated	0.42	0.68	0.68	WGCNA on raw counts vs. the agent’s normalised counts
b-8-4	Agent	0.00	0.00	0.00	ElasticNet without feature scaling
a-9-4	Agent	0.09	0.09	0.09	xCell rnaseq=FALSE ; default run matched but was not submitted
a-5-10	Agent	0.10	0.55	0.55	Leiden resolution 1.0 vs. 0.5 (55% agreement after relabelling)
a-11-7	Agent	0.22	0.32	0.03	Correlated with raw instead of inverted essentiality
a-6-2	Agent	0.30	0.19	0.21	Custom DADA2 instead of nf-core/ampliseq
a-15-7	Agent	0.45	0.49	0.50 [†]	Shortened iClusterPlus λ grid, different seed
a-6-4	Agent	0.45	0.44	0.54	DESeq2 contrast reversed, poscounts
a-2-5	Agent	0.48	0.48	0.72	All gene pairs drawn instead of pairs seen in ≥ 2 samples

*0.499998. [†]0.496.

Table S3: Claude-with-skills tasks whose pass/fail differs between the two judges under the fixed evaluator.

Task	GPT judge	Claude judge	Why they differ
a-14-4	0.42	0.82	Image-judge prompt now treats a vertically mirrored heatmap as the same data
a-3-6	0.42	0.75	Trailing # lines in STAR 2.7.11 junction files dropped before the Claude run; rows match
a-2-5	0.48	0.72	Image judge gave more partial credit to the circos plot
a-6-4	0.44	0.54	Table re-plan and key normalisation; the DESeq2 contrast is still reversed
a-9-6	0.55	0.45	Image judge near the cut-off; underlying cause is the reference’s contig counting
b-3-8	0.56	0.40	Image judge near the cut-off; underlying cause is the stale reference image

Table S4: Computational cost per task. Specialist-agent figures are from the PromptBio-Bench release. Claude token counts are summed over all API calls in the subagent transcripts and are dominated by cached prompt re-reads, which are billed at a fraction of the input price; they are therefore not directly comparable with the specialists’ input tokens. Output tokens are not reliably recorded in the transcripts; for the skills run they were estimated at about 1.35 M in total from context growth.

Agent	Wall-clock time (s)			Tokens per task (mean)	
	Mean	Median	Max	Input	Output
Biomni	171	82	3,600	98.6 k	4.6 k
STELLA	275	74	3,600	1,294 k	23.7 k
ToolsGenie	218	100	3,600	59.4 k	7.0 k
Claude Opus 5.5 + skills	171	39	2,990	414 k (90% cache reads)	≈5.5 k
Claude Opus 5.5, no skills	93	31	1,837	306 k (92% cache reads)	≈5.5 k

Table S5: Per-task similarity scores. Bold marks a pass (similarity ≥ 0.5 , judged on the unrounded score; e.g. b-4-15's 0.499998 prints as 0.50 but fails). Difficulty: L/M/H. Original: original evaluator (specialists: GPT-5.4 judge; Claude: GPT-5.6/5.6-sol). Fixed: fixed evaluator with the GPT-5.6-sol/6.1-sol (GPT) or Claude Opus 5.5 (Claude) judge. Also available as `supplementary_table_S5_scores.csv`.

Task	Diff.	Original evaluator				Skills, fixed		No skills, fixed	
		Biomni	STELLA	ToolsGenie	Claude+skills	GPT	Claude	GPT	Claude
a-1-1	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-1-2	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-1-3	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-1-4	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-1-5	L	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-1-6	L	0.75	0.75	0.75	1.00	1.00	1.00	1.00	1.00
a-1-7	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-1-8	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-1-9	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-1-10	M	0.77	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-1-11	L	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
a-1-12	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-1-13	M	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
a-1-14	M	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
a-1-15	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-2-1	L	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
a-2-3	M	0.53	0.99	0.87	0.96	0.96	0.96	0.96	0.96
a-2-5	M	0.72	0.32	0.26	0.48	0.48	0.72	0.18	0.72
a-2-6	M	0.00	0.93	0.00	0.53	0.75	0.75	0.75	0.75
a-2-7	L	0.88	0.92	0.82	0.98	0.96	0.88	0.94	0.90
a-2-8	M	0.21	0.28	0.72	0.75	0.62	0.98	0.33	0.99
a-2-9	M	1.00	1.00	0.08	1.00	1.00	1.00	1.00	1.00
a-2-10	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-2-13	M	0.12	0.50	0.50	0.50	0.50	0.50	0.50	0.50
a-2-15	H	0.94	0.90	0.90	0.98	0.98	0.95	0.98	0.95
a-3-1	M	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
a-3-2	M	0.03	0.67	1.00	0.56	0.56	0.56	0.68	0.68
a-3-3	M	0.48	1.00	0.00	1.00	1.00	1.00	1.00	1.00
a-3-4	M	0.44	0.43	0.48	0.98	0.97	0.96	0.98	0.95
a-3-5	H	0.18	0.00	0.32	0.33	1.00	1.00	1.00	1.00
a-3-6	H	0.00	0.00	0.00	0.00	0.42	0.75	0.00	0.76
a-3-7	L	0.43	0.54	0.43	0.54	0.54	0.54	0.54	0.54
a-3-8	M	0.02	0.88	1.00	0.83	0.61	0.89	0.95	0.95
a-3-9	H	1.00	0.00	1.00	1.00	1.00	1.00	0.88	0.88
a-3-10	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-3-11	M	0.42	0.85	0.38	1.00	1.00	1.00	1.00	1.00
a-3-12	M	0.00	0.00	0.00	0.76	0.82	0.87	0.99	0.99
a-3-13	H	0.00	0.00	0.77	0.97	0.96	0.96	0.95	0.95
a-3-14	M	1.00	0.62	0.00	1.00	1.00	1.00	1.00	1.00
a-3-15	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-4-1	M	0.74	1.00	0.20	1.00	1.00	1.00	1.00	1.00
a-4-2	M	0.00	1.00	0.22	1.00	1.00	1.00	0.78	0.78
a-4-4	M	0.08	0.03	0.08	0.00	0.10	0.15	0.29	0.45
a-4-5	L	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
a-4-6	H	0.00	0.52	0.00	0.27	0.27	0.27	0.27	0.27
a-4-7	M	0.00	0.00	0.00	0.99	0.99	0.99	1.00	1.00
a-4-8	H	0.58	0.58	0.42	0.76	0.82	0.78	0.76	0.70
a-4-9	M	1.00	0.00	1.00	0.81	0.94	0.94	1.00	1.00
a-4-10	M	0.74	0.94	1.00	1.00	1.00	1.00	1.00	1.00
a-5-1	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-5-2	M	0.72	0.42	0.76	0.82	0.88	0.80	0.82	0.85
a-5-3	H	0.58	0.26	0.58	0.78	0.86	0.72	0.86	0.75

Task	Diff.	Biomni	STELLA	ToolsGenie	Claude+skills	GPT	Claude	GPT	Claude
a-5-4	M	0.00	0.95	0.08	0.95	0.95	0.95	0.42	0.42
a-5-5	M	0.74	0.65	0.18	0.65	0.65	0.65	0.65	0.65
a-5-6	M	0.62	0.65	0.58	0.56	0.56	0.56	0.38	0.56
a-5-7	M	0.67	1.00	0.76	0.86	0.89	0.90	0.83	0.90
a-5-8	M	0.88	0.00	0.87	0.90	0.92	0.85	0.93	0.85
a-5-9	M	0.62	0.00	0.57	0.70	0.69	0.62	0.63	0.62
a-5-10	H	0.93	0.95	0.91	0.10	0.55	0.55	0.74	0.74
a-6-1	M	0.77	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-6-2	M	0.10	0.50	0.00	0.30	0.19	0.21	0.20	0.20
a-6-3	L	1.00	0.00	1.00	0.00	1.00	1.00	1.00	1.00
a-6-4	M	0.20	0.24	0.28	0.45	0.44	0.54	0.30	0.55
a-6-5	M	0.50	0.54	0.45	0.61	0.51	0.51	0.51	0.51
a-6-6	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-6-7	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-6-8	L	1.00	1.00	1.00	0.97	0.82	0.80	0.82	0.80
a-6-9	M	0.27	0.60	0.00	0.60	0.60	0.60	0.83	0.83
a-6-10	M	0.06	1.00	1.00	0.60	0.81	0.81	0.73	0.81
a-7-1	L	0.00	0.00	0.75	0.59	0.58	0.67	0.78	0.59
a-7-3	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-7-4	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-7-5	M	0.94	0.94	0.94	0.97	0.99	0.96	0.99	0.95
a-7-6	L	0.56	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-7-7	H	0.00	0.00	0.00	0.00	0.04	0.04	0.08	0.08
a-7-8	H	0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00
a-7-9	H	0.77	0.38	0.43	0.80	0.84	0.72	0.83	0.73
a-7-10	M	0.89	0.83	0.90	0.98	0.99	0.96	0.98	0.96
a-8-1	M	0.67	0.83	1.00	0.60	1.00	1.00	0.00	0.00
a-8-2	M	0.25	0.33	0.22	1.00	1.00	1.00	1.00	1.00
a-8-3	M	0.68	0.72	0.80	0.92	0.62	0.80	0.62	0.75
a-8-4	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-8-5	M	0.50	0.00	0.50	0.50	0.50	0.50	0.50	0.50
a-9-1	H	0.56	0.49	0.00	0.58	0.58	0.58	0.41	0.41
a-9-2	M	0.06	0.48	0.00	0.91	0.90	0.88	0.89	0.88
a-9-3	M	0.00	0.17	0.17	1.00	1.00	1.00	1.00	1.00
a-9-4	M	0.00	0.00	0.69	0.09	0.09	0.09	0.09	0.09
a-9-5	M	0.00	0.00	0.00	0.85	0.85	0.85	0.85	0.85
a-9-6	M	0.84	0.72	0.22	0.65	0.55	0.45	0.55	0.45
a-9-7	M	1.00	1.00	1.00	0.86	1.00	0.86	1.00	0.86
a-9-9	M	0.54	1.00	0.54	0.00	0.00	0.00	0.00	0.00
a-10-1	L	1.00	0.67	1.00	1.00	1.00	1.00	1.00	1.00
a-10-2	M	0.36	0.00	0.98	0.14	0.12	0.07	0.14	0.07
a-10-3	H	0.67	0.00	1.00	0.67	0.67	0.67	0.67	0.67
a-10-4	M	0.97	0.98	0.97	0.92	0.98	0.97	0.99	0.97
a-10-5	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-11-1	M	0.86	0.79	0.65	0.59	0.59	0.57	0.61	0.57
a-11-2	H	0.75	0.52	0.30	0.42	0.68	0.68	0.64	0.64
a-11-3	M	0.22	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-11-4	M	0.62	0.63	0.57	0.99	0.99	0.99	0.96	0.99
a-11-5	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-11-6	M	0.08	0.08	0.08	0.12	0.12	0.10	0.12	0.15
a-11-7	M	0.37	0.38	0.04	0.22	0.32	0.03	0.32	0.05
a-11-8	H	0.86	0.64	0.81	0.98	0.98	0.93	0.98	0.94
a-11-9	M	0.65	0.71	0.66	0.77	0.69	0.77	0.74	0.79
a-11-10	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-12-1	M	0.20	0.98	0.96	0.67	0.75	0.69	0.72	0.73
a-12-2	M	0.08	0.94	0.90	0.78	0.62	0.50	0.68	0.55
a-12-3	L	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00
a-12-4	M	0.60	0.00	0.46	1.00	1.00	1.00	1.00	1.00
a-12-5	M	0.90	0.95	0.95	0.98	0.98	0.93	0.98	0.93

Task	Diff.	Biomni	STELLA	ToolsGenie	Claude+skills	GPT	Claude	GPT	Claude
a-13-1	M	0.00	0.00	0.66	1.00	1.00	1.00	1.00	1.00
a-13-2	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-13-3	H	0.14	0.09	0.86	1.00	1.00	1.00	1.00	1.00
a-13-4	H	0.81	0.81	1.00	0.81	1.00	0.95	1.00	0.95
a-13-5	H	0.98	0.64	0.99	1.00	1.00	1.00	1.00	1.00
a-14-1	H	0.30	0.00	0.11	0.65	0.65	0.63	0.65	0.63
a-14-2	H	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
a-14-3	M	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
a-14-4	M	0.72	0.76	0.98	0.96	0.42	0.82	0.48	0.65
a-14-5	M	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
a-15-1	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-15-2	H	0.27	1.00	0.00	1.00	1.00	1.00	1.00	1.00
a-15-3	H	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
a-15-4	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
a-15-5	M	0.57	0.57	1.00	0.57	0.57	0.57	0.57	0.57
a-15-6	H	0.97	0.89	0.97	0.89	0.97	1.00	1.00	1.00
a-15-7	H	0.74	0.69	0.62	0.45	0.49	0.50	0.48	0.55
a-15-8	H	0.42	0.00	0.00	0.73	0.76	0.80	0.66	0.75
a-15-9	H	0.42	0.32	0.43	0.36	0.20	0.12	0.27	0.15
a-15-10	H	0.34	0.00	1.00	1.00	1.00	1.00	1.00	0.94
b-1-1	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-1-2	L	0.99	1.00	0.97	1.00	1.00	1.00	1.00	1.00
b-1-3	L	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00
b-1-4	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-1-5	L	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00
b-2-1	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-2-2	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-2-3	M	1.00	1.00	1.00	0.80	0.99	0.99	0.99	0.99
b-2-4	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-2-5	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-2-6	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-2-7	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-2-8	L	0.50	0.00	0.50	1.00	1.00	1.00	1.00	1.00
b-2-9	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-2-10	M	0.94	0.00	0.94	0.96	1.00	1.00	1.00	1.00
b-2-11	M	1.00	0.97	1.00	0.59	0.59	0.65	0.59	0.75
b-2-12	L	1.00	1.00	0.96	1.00	1.00	1.00	1.00	1.00
b-2-13	M	0.49	1.00	0.49	0.48	0.96	0.92	1.00	1.00
b-2-14	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-2-15	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-3-1	M	0.96	0.95	0.95	0.99	0.99	0.95	0.99	0.95
b-3-2	L	0.97	0.96	0.96	0.98	0.99	0.96	0.99	0.96
b-3-3	L	0.96	0.88	0.95	0.99	1.00	0.97	1.00	0.97
b-3-4	L	0.98	0.97	0.98	1.00	1.00	0.97	1.00	0.97
b-3-5	M	0.99	0.98	0.97	0.99	1.00	0.97	0.99	0.96
b-3-6	M	0.97	0.97	0.96	0.99	0.99	0.98	0.99	0.98
b-3-7	L	0.93	0.00	0.94	0.98	0.99	0.95	0.98	0.92
b-3-8	M	0.88	0.72	0.84	0.58	0.56	0.40	0.58	0.40
b-3-9	M	0.72	0.68	0.42	0.99	1.00	0.95	0.99	0.93
b-3-10	M	0.93	0.00	0.90	0.97	0.98	0.95	0.97	0.93
b-3-11	M	0.92	0.00	0.93	0.96	0.97	0.93	0.94	0.92
b-3-12	M	0.72	0.82	0.88	0.97	0.98	0.93	0.98	0.93
b-3-13	M	0.76	0.74	0.26	0.82	0.86	0.85	0.84	0.85
b-3-14	M	0.95	0.90	0.95	0.98	0.99	0.93	0.99	0.92
b-3-15	L	0.56	0.66	0.62	0.46	0.32	0.30	0.40	0.35
b-4-1	M	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
b-4-2	M	0.90	0.94	0.90	0.98	0.99	0.96	0.99	0.95
b-4-3	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-4-4	L	1.00	1.00	0.25	1.00	1.00	1.00	1.00	1.00

Task	Diff.	Biomni	STELLA	ToolsGenie	Claude+skills	GPT	Claude	GPT	Claude
b-4-5	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-4-6	M	0.67	1.00	1.00	0.58	1.00	1.00	1.00	1.00
b-4-7	L	1.00	0.00	1.00	1.00	1.00	1.00	0.98	0.98
b-4-8	M	0.90	1.00	0.67	1.00	1.00	1.00	1.00	1.00
b-4-9	M	1.00	0.69	1.00	0.59	0.98	0.98	1.00	1.00
b-4-10	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-4-11	M	0.97	0.97	0.98	0.99	1.00	0.98	1.00	0.99
b-4-12	M	1.00	0.68	0.75	0.83	0.83	0.83	0.17	0.75
b-4-13	H	1.00	0.76	1.00	0.61	0.64	0.64	0.64	0.32
b-4-14	L	0.99	0.99	1.00	0.99	0.99	0.99	0.99	0.99
b-4-15	L	0.50	0.50	0.50	0.50	1.00	1.00	1.00	1.00
b-5-1	L	0.96	0.00	0.78	0.99	0.99	0.96	0.99	0.95
b-5-2	L	1.00	1.00	0.18	1.00	1.00	1.00	1.00	1.00
b-5-3	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-5-4	M	0.95	0.00	0.95	0.96	0.98	0.96	0.98	0.95
b-5-5	M	0.64	0.70	0.82	0.91	0.83	0.83	0.89	0.95
b-5-6	M	0.95	0.99	0.97	1.00	1.00	0.99	1.00	0.99
b-5-7	M	0.98	0.98	0.92	0.97	0.98	0.96	0.98	0.97
b-5-8	L	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-5-9	M	0.99	0.99	0.49	0.85	0.94	0.94	0.94	0.94
b-5-10	H	0.82	0.99	0.81	1.00	1.00	1.00	1.00	1.00
b-6-1	L	1.00	1.00	1.00	0.58	1.00	1.00	1.00	1.00
b-6-2	L	0.96	0.68	0.62	0.99	0.99	0.97	0.99	0.97
b-6-3	M	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
b-6-4	M	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00
b-6-5	M	0.82	0.56	0.62	0.86	0.70	0.60	0.78	0.70
b-6-6	M	0.00	0.00	0.78	0.72	0.88	0.78	0.78	0.75
b-6-7	M	0.60	0.50	0.53	0.50	0.53	0.62	0.53	0.62
b-6-8	H	0.98	0.85	0.98	0.95	0.95	0.95	0.97	0.97
b-6-9	M	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
b-6-10	M	0.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00
b-7-1	M	0.72	0.62	0.60	0.74	0.95	0.95	0.94	0.94
b-7-2	M	0.61	0.72	0.65	0.62	0.96	0.94	0.85	0.83
b-7-3	H	0.73	0.59	0.65	0.57	0.72	0.74	0.71	0.70
b-7-4	M	0.50	0.50	0.50	0.24	0.99	0.99	0.99	0.99
b-7-5	L	0.57	0.50	0.50	0.46	0.98	0.98	0.98	0.98
b-7-6	L	1.00	0.00	0.75	0.88	1.00	1.00	1.00	1.00
b-7-7	M	0.56	0.58	0.56	0.56	0.82	0.82	0.43	0.43
b-7-8	M	0.62	0.50	0.57	0.58	1.00	1.00	1.00	1.00
b-7-9	M	0.88	0.78	0.68	0.76	1.00	1.00	1.00	1.00
b-7-10	H	0.57	0.00	0.56	0.55	0.75	0.75	0.74	0.74
b-8-1	M	0.96	1.00	1.00	0.89	0.96	0.96	0.96	0.96
b-8-2	M	0.83	0.83	0.98	1.00	1.00	1.00	0.60	0.60
b-8-3	M	0.60	0.60	0.60	0.67	0.51	0.51	0.51	0.51
b-8-4	M	0.67	0.67	0.67	0.00	0.00	0.00	0.00	0.00
b-8-5	H	0.50	0.50	0.50	0.50	0.99	0.99	0.99	0.99
b-8-6	M	0.75	0.75	0.75	0.53	0.99	0.99	0.99	0.99
b-8-7	H	0.57	0.62	0.57	0.62	0.20	0.21	0.21	0.21
b-8-8	H	1.00	0.75	0.82	1.00	1.00	1.00	1.00	1.00
b-8-9	M	0.50	0.50	0.50	0.57	0.53	1.00	0.50	1.00
b-8-10	H	1.00	0.00	1.00	0.73	1.00	1.00	1.00	1.00
b-9-2	M	0.00	0.00	1.00	0.75	1.00	1.00	1.00	1.00
b-9-3	M	1.00	1.00	0.99	0.81	1.00	1.00	1.00	1.00
b-9-4	H	0.59	0.58	0.55	0.51	0.29	0.30	0.29	0.30
b-10-1	M	0.75	0.75	0.75	0.69	1.00	1.00	1.00	1.00
b-10-2	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-10-3	L	0.96	1.00	0.96	1.00	1.00	1.00	1.00	1.00
b-10-4	H	0.85	0.53	0.85	0.82	0.97	0.97	0.97	0.97
b-10-5	H	0.91	0.91	0.91	0.79	1.00	1.00	1.00	1.00

Task	Diff.	Biomni	STELLA	ToolsGenie	Claude+skills	GPT	Claude	GPT	Claude
b-10-6	M	0.82	0.63	0.82	0.66	1.00	1.00	1.00	1.00
b-10-7	M	0.57	0.82	0.57	0.57	1.00	1.00	1.00	1.00
b-10-8	L	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
b-10-9	M	0.50	0.75	0.00	0.25	1.00	1.00	0.50	1.00
b-10-10	M	0.75	0.00	0.00	1.00	1.00	1.00	1.00	1.00
b-11-1	M	0.99	0.96	0.99	1.00	1.00	1.00	1.00	1.00
b-11-2	M	0.99	0.00	0.98	0.99	1.00	0.97	1.00	0.97
b-11-3	M	0.96	0.54	0.83	0.64	0.50	0.98	0.50	0.98
b-11-4	H	0.93	0.80	0.96	0.49	1.00	1.00	1.00	1.00
b-11-5	H	0.37	0.46	0.73	0.56	0.58	0.86	0.55	0.85
b-12-1	M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-12-2	H	0.95	0.00	0.96	0.93	0.93	0.93	0.93	0.93
b-12-3	H	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-12-4	H	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
b-12-5	M	0.94	1.00	1.00	1.00	1.00	1.00	1.00	1.00