



INTUITIONLABS / RESEARCH & PRACTICAL GUIDANCE

Claude Programmatic Tool Calling in Bioinformatics: Test Plan

Build practical AI for pharma and biotech with IntuitionLabs. We help life-science teams turn complex information and workflows into useful software, governed knowledge systems and AI tools.

Published by IntuitionLabs · 2026-10-27 · claude programmatic tool calling · bioinformatics · allowed_callers · code execution tool · zero data retention · llm tool calling validation · computer software assurance · gxp data integrity · ai scientific workflow automation · tool-call ledger

Original article: <https://intuitionlabs.ai/articles/claude-programmatic-tool-calling-bioinformatics-test-plan>



In this article

1. [Executive Summary](#)
 2. [Introduction and Background](#)
 3. [Key Changes: The June 2026 Capability and Its Constraints](#)
 4. [Architecture, Fit, and Data Classification](#)
 5. [Tool Contracts and Execution Controls](#)
 6. [Qualification: A/B Protocol, Evidence Pack, and Decision Matrix](#)
 7. [Data Analysis and Evidence](#)
 8. [Case Studies and Real-World Examples](#)
 9. [Implications and Future Directions](#)
 10. [Frequently Asked Questions \(FAQs\)](#)
 11. [Conclusion](#)
-

Executive Summary

Programmatic tool calling (PTC) changes where intermediate results live. Claude writes Python, a sandboxed code container runs it, and each call to a permitted tool is paused, handed to the client application, answered by the client, and resumed. The intermediate results are not loaded into the model's context window ([platform.claude.com](#)), so only the final code output returns to Claude. This is a different execution topology from classic tool use, and it deserves its own qualification rather than reuse of an MCP (Model Context Protocol) or direct tool-calling approval.

The release timeline matters. Programmatic tool calling launched in beta on November 24, 2025 ([platform.claude.com](#)). On June 18, 2026, the Python, TypeScript, Go, Java, Ruby, PHP, and C# SDKs added support for the `code_execution_20260120` tool version that the feature requires ([platform.claude.com](#)). The same June 18 entry says no beta header is required and lists availability on Claude Fable 5 and Claude Mythos 5, among other models.

The hidden tradeoff is retention. Anthropic's data-retention table lists programmatic tool calling as not eligible for zero data retention (ZDR), because it is built on code execution containers with data retained up to 30 days ([platform.claude.com](#)). The programmatic tool calling page repeats that container data, including execution artifacts and outputs, is retained for up to 30 days. Any workload that depends on ZDR, or on contractual terms that exclude retained container artifacts, should be treated as a reject or redesign case until a privacy lead signs off.

Anthropic's efficiency numbers are vendor measurements on named workloads, not forecasts for pharma. A 75-tool project-management benchmark showed roughly 38 percent fewer billed input tokens with no change in task accuracy ([platform.claude.com](#)). A sequential benchmark with one or two calls per turn cost roughly 8 percent more ([platform.claude.com](#)). Production traffic with 10 to 49 tool definitions saw typical savings of 20 to 40 percent ([platform.claude.com](#)), and web research benchmarks improved performance by an average of 11 percent with 24 percent fewer input tokens ([platform.claude.com](#)).

The recommended path is an A/B qualification: run representative, non-sensitive workflows through both direct and programmatic calling, compute percentage change as (programmatic minus direct) divided by direct times 100 from measured runs only, and score accuracy, provenance completeness, tool failures, and human-review time alongside tokens. For regulated contexts, the FDA's computer software assurance (CSA) guidance applies directly to specified medical-device production and quality management system software (www.fda.gov), while the UK MHRA GxP data integrity guidance is a broader reference for governance, metadata, and audit trails (www.gov.uk). This report supplies the scorecard, gates, contract, test matrix, and decision matrix for that qualification.

Introduction and Background

Life-science platform teams now routinely connect language models to literature indexes, sequence databases, trial registries, and electronic lab notebooks. Anthropic's own life-sciences materials describe connectors for PubMed, 10x Genomics, and other services (www.anthropic.com), and its Bio Research plugin page says it consolidates 10 MCP server integrations and 6 analysis skills (claude.com). The plugin's public GitHub README gives different counts, 11 MCP server integrations and 5 analysis skills (github.com), so the two first-party descriptions may reflect different versions and should be re-checked at adoption time.

Workflows on that surface include single-cell RNA-seq quality control (claude.com) and nf-core pipelines for RNA-seq, variant calling, and ATAC-seq (claude.com), which tend to produce many calls with large intermediate results. In classic tool use, every result passes through the model's context. PTC lets the model write a script that loops, filters, and aggregates, and return only a summary.

This report does not re-explain MCP, Claude Skills, or generic tool use. It analyzes one question: when does the code-container topology make sense for a scientific workflow, which data and tools may be exposed to it, and how can a team prove any efficiency gain without weakening provenance or controls? Public search-demand estimates for Claude-in-life-sciences terms are unavailable, so demand is treated as unknown rather than zero.

This report is limited to prose guidance on qualification design. This report is not legal, regulatory, or security advice; consult quality, privacy, and legal professionals before production decisions. All values are dated as of October 2026 unless stated otherwise.

Key Changes: The June 2026 Capability and Its Constraints

What changed on June 18, 2026

The June 18, 2026 release-note entry is narrow but operationally important. SDKs in seven languages gained support for `code_execution_20260120`, and the tool needs no `anthropic-beta` header across its three current versions (platform.claude.com). That version adds REPL (read-eval-print loop) state persistence along with PTC. A later variant, `code_execution_20260521`, runs the same runtime and only discloses the per-cell time limit to the model. Both strings are accepted and interchangeable in `allowed_callers`, so teams should record which one they send.

The execution loop as a different topology

In classic tool use, the model emits a tool call, the client runs it, and the result returns to the model's context. In PTC, Claude writes Python that calls the tool as a function. Tools opted in appear to the code as async Python functions, so the script can fan out calls in parallel. When the code calls one, execution pauses and the API returns a tool-use block to the client. The client then supplies each result, and the final code output is the only thing that reaches the model.

The control surface is the `allowed_callers` field on each tool definition. Anthropic documents three behaviors:

- **Direct only (default):** Claude is guided to call the tool directly.
- **Code execution only:** Claude is guided to call the tool only from within code execution.
- **Both:** Claude may choose either route, although the documentation recommends picking one mode per tool for clearer guidance.

The critical caveat for security and validation teams is that `allowed_callers` is guidance, not enforcement. The page says it is not a hard API-level block on direct calls, and it warns against relying on the field as a security boundary. That is consistent with OWASP's advice to implement authorization in downstream systems (genai.owasp.org). Every tool-use block carries a caller field, and the programmatic caller's identifier is the ID of the code-execution block that made the call. That field is the natural join key for a tool-call ledger: the client's own logs can record, for each call, whether it was direct or programmatic and which code run produced it.

Retention, zero data retention, and platform availability

The documented retention position is the most consequential constraint for life-science use. The ZDR eligibility table lists PTC as built on code execution containers with data retained up to 30 days, and it explains that code execution runs in persistent containers (platform.claude.com). The build-overview feature table also marks the feature "Not ZDR eligible". By contrast, under ZDR proper, Anthropic does not store customer prompts or responses at rest after the response is returned. Using a non-eligible feature therefore steps outside that guarantee for the data it touches.

Container lifetime has several clocks. A new container is created per request unless the caller reuses one. While programmatic calls are pending, the container ID is required on the next request. After about 5 minutes of inactivity a container is checkpointed, and a request that carries its ID inside the 30-day window restores it (platform.claude.com), so the 5-minute value does not shorten the 30-day data retention, and no container can be reused more than 30 days after creation. Separately, a pending programmatic tool call times out after about 4 minutes (platform.claude.com). Anthropic's wording on the idle period includes the word "currently", so the 5-minute value should be re-verified in each qualification cycle.

Billing, rate limits, and unsupported combinations

PTC uses the same pricing as code execution. Tool results from programmatic calls do not count toward input or output token usage; only the final code-execution result and Claude's response do. Rate limits are not relaxed: each call from code counts individually, and NCBI, for example, recommends no more than three URL requests per second (www.ncbi.nlm.nih.gov), so the broker, not the model, should throttle per service. Code

execution is billed at \$0.05 per hour per container once an organization exceeds its free allotment (platform.claude.com), and each organization receives 1,550 free hours per month (platform.claude.com). Execution time carries a 5-minute minimum, and files preloaded onto a container are billed even if the tool is never called. For most single-lab pilots the container charge will be small relative to token charges, but the free-hour pool is organization-wide, so enterprise-scale batch runs should model it.

Table 1 below summarizes the compatibility facts a platform team needs, each date-stamped to the October 2026 reading.

Attribute (as of Oct 2026)	Documented value	Qualification implication
Minimum code tool version	code_execution_20260120 or later; SDK support added June 18, 2026	Pin and record the version string in change control.
Beta header	None required for the three current versions	No beta-flag audit needed, but version drift is still possible.
ZDR eligibility	Not eligible; data retained up to 30 days	Blocks ZDR-dependent workloads.
Container lifetime	Idle containers checkpointed after about 5 minutes and restorable by ID; maximum reuse 30 days (the 5 minutes does not shorten retention)	Design for expired-container recovery.
Pending-call timeout	About 4 minutes; 90-second per-cell wall clock	Tool brokers must answer within the window.
Rate limits	Same as regular tool calls; each call counts	Add broker-side throttling per upstream service.
Unsupported	Strict tools; forcing via tool_choice; MCP connector tools (platform.claude.com)	Schema enforcement and remote MCP routes need redesign.
Sandbox network	Internet access disabled; packages cannot be installed at runtime (platform.claude.com)	The container cannot reach databases itself; the client must.

The table shows a recurring pattern: most limits push responsibility to the client. Because the container has no network, the broker that fulfills tool calls is where authorization, throttling, and logging must live. The exclusion of MCP connector tools also means that a remote MCP server attached through the connector, which allows direct connection to remote servers from the Messages API, cannot simply be reused; teams with internal MCP services would need client-side tool definitions that wrap them. The MCP connector documentation also notes that local STDIO servers cannot be connected directly.

Architecture, Fit, and Data Classification

Reference architecture

The topology can be described as a seven-step flow:

1. **Model context:** Claude receives the task and tool descriptions, and writes Python.

2. **Code container:** the script runs in Anthropic's sandbox, which uses the same sandbox containers as code execution, with one CPU, file access limited to the workspace directory, and no internet.
3. **Pause and handoff:** each tool call returns to the client as a tool-use block marked with a caller field.
4. **Tool broker (client-owned):** validates the call against an allow-list, applies rate limits, injects credentials the model never sees, and invokes the internal database or API.
5. **Result validation:** the broker checks schema, size, and content before returning a text result.
6. **Ledger and logs:** the broker records the call, parameters, result checksum, and caller ID. Anthropic recommends logging all tool calls and results.
7. **Final response:** the code's final output, not the intermediates, enters context.

Strong-fit scientific patterns

Anthropic lists fan-out or parallel operations across many items as a strong fit, along with agentic search and retrieval where iterative querying and result filtering dominate. Mapped to life-science work, four patterns qualify for pilot:

- **Parallel database lookups:** resolving a few hundred gene or accession identifiers against several reference services, then returning only discrepancies.
- **Batch sample-metadata checks:** comparing a sample sheet against a LIMS (laboratory information management system) export and returning only mismatched rows.
- **Evidence retrieval:** querying literature and trial registries, deduplicating, and returning a ranked table. A ClinicalTrials.gov connector covering 500,000+ studies (support.claude.com) is the kind of large result space where filtering in code pays off.
- **Large-result filtering:** Anthropic's own worked example reduced 200KB of raw expense data to 1KB of results (www.anthropic.com), and its MCP post describes an agent seeing five rows rather than 10,000 (www.anthropic.com).

Weak-fit patterns

Anthropic is equally direct about weak fits: strictly sequential workflows where each call depends on reasoning over the previous result, few calls with small responses where overhead can exceed savings, and tools that require immediate user feedback between calls. In scientific terms, expect weak fit for interpretive triage (a variant-classification step that depends on the prior annotation), single-record lookups, and interactive approval steps such as a reviewer confirming a protocol before a write. Any workflow that must run under ZDR is a fourth weak-fit class by constraint, not by performance.

Workflow-fit scorecard

A simple scorecard can rank candidates before any test spend. Score each factor from 0 (poor fit) to 2 (strong fit):

- **Number of tools:** 1 to 2 tools scores 0; 3 to 9 scores 1; 10 or more scores 2.
- **Degree of parallelism:** strictly sequential scores 0; partially parallel scores 1; independent fan-out scores 2.

- **Intermediate result size and reasoning between calls:** small results or per-step interpretation score 0; large tabular results the model need not read, with no interpretation between calls, score 2.
- **Data class:** any class barred from retained containers scores a hard stop, not a number.
- **Reproducibility need:** score 0 where deterministic replay is impossible, and fix that gap before pilot.

Data classification gate

Retention of container artifacts for up to 30 days makes the data class the first gate. The gate below uses five tiers and explicitly prohibits universal conclusions: a class that is acceptable at one institution may be barred at another by contract, consent, or law.

- **Public data:** accessions, published literature, open reference databases. Generally the lowest-risk class, subject to the license of each source.
- **Proprietary assay data:** unpublished results and compound data. Acceptable only if the organization's agreements permit 30-day vendor retention of derived artifacts.
- **Coded subject data:** records keyed by a code with the key held elsewhere. The EDPB treats pseudonymised data that can be attributed with additional information as information on an identifiable natural person (www.edpb.europa.eu), and the UK ICO states pseudonymised data remains in scope of data protection law (ico.org.uk). Under HIPAA guidance, disclosing a re-identification code is itself considered a disclosure of protected health information (www.hhs.gov).
- **Direct identifiers:** names, contact details, and similar fields. Exclude by default from any retained container.
- **Regulated records:** data under a data use certification, special-category rules, or GxP record requirements. GDPR Article 9 places genetic data among special categories (gdpr-info.eu), and Article 4 defines genetic data as personal data about inherited or acquired genetic characteristics (gdpr-info.eu).

Controlled-access genomic data adds contractual duties. NIH Security Best Practices say all users in possession of NIH controlled-access data must protect it under NIST SP 800-171 (www.grants.nih.gov), with updated user standards applying effective February 25, 2026 (grants.nih.gov), and an NIH notice addresses sharing such data with public generative AI tools through prompts (grants.nih.gov). Whether an enterprise API arrangement falls within those provisions is a question for the data access committee and counsel, not for this report.

Tool Contracts and Execution Controls

The tool-result contract

Tool results return to the code as strings, and Anthropic advises documenting the output format in detail so Claude can deserialize it in code. The contract should therefore be explicit, versioned JSON. The minimum fields:

- **Schema version:** a semantic version string, so a stale schema is detectable.

- **Source identifier:** the source URL or accession. NCBI, for example, uses the accession.version identifier rather than the older GI number (www.ncbi.nlm.nih.gov), which is the kind of versioned identifier a contract should require.
- **Query time and database release:** UniProt release notes date-stamp each release, such as the release dated 02-Sep-2026 (ftp.uniprot.org), and Ensembl's 116 release notes say it is the final release on the current platforms (www.ebi.ac.uk), with the current platforms remaining available via archives (www.ebi.ac.uk). NCBI's EInfo returns the date of the last update of a database (www.ncbi.nlm.nih.gov). Capture whichever release marker the upstream service exposes.
- **Filters and warnings:** the exact query parameters and any truncation or deprecation warnings.
- **Record count and checksum:** a count and a hash of the canonical payload, so a partial or oversized result is detectable.
- **Limits and idempotency:** row and byte caps, a timeout budget, an idempotency key, and an allow-listed operation name.

This contract also serves provenance. The FAIR principles ask that metadata be associated with detailed provenance (www.nature.com), and Workflow Run RO-Crate is implemented by several workflow management systems (arxiv.org), with its Provenance Run Crate profile offering the highest level of granularity (www.ncbi.nlm.nih.gov). A pipeline engine such as Nextflow allows self-contained pipelines with managed versions (www.nextflow.io), which gives an A/B harness a reproducible baseline.

Injection and execution controls

Tool results are an untrusted channel. Anthropic warns to be aware of code injection risks if the output is interpreted or run as code. NIST explains that indirect prompt injection arises because generative models combine data and instruction channels (nvlpubs.nist.gov) and notes that it can hijack an agent into an attacker-specified task (nvlpubs.nist.gov). NIST's generative AI profile describes it as injecting prompts into data likely to be retrieved (nvlpubs.nist.gov), which is exactly what a database free-text field is. OWASP adds that injected prompts need not be human-visible (genai.owasp.org) and that it is unclear whether fool-proof prevention exists (genai.owasp.org), so mitigations should limit impact.

Controls that follow from this evidence:

- **Never interpret strings as code:** no eval, exec, or shell calls on tool output. OWASP advises treating the model like any other user under zero trust (genai.owasp.org) and avoiding open-ended extensions such as running a shell command (genai.owasp.org).
- **Enforce authorization downstream:** authorization belongs in the downstream system rather than the model's decision (genai.owasp.org). Because allowed_callers is not a boundary, the broker must reject disallowed operations regardless of caller.
- **Isolate credentials:** keep secrets in the broker. OWASP recommends handling privileged functions in code rather than giving them to the model (genai.owasp.org). The MHRA guidance adds that shared or generic logins should not be used for systems that handle GxP data (assets.publishing.service.gov.uk).
- **Restrict network and filesystem:** the managed container has no internet; the broker's egress should be equally restricted.

- **Human approval for privileged writes:** OWASP recommends human-in-the-loop controls for privileged operations (genai.owasp.org). Keep writes, deletions, and submissions out of the code-callable tool set.
- **Validate before returning:** check schema, size, and a deny-pattern scan on free-text fields before the broker sends any result into the container.

Table 2 below turns these controls into a negative-test matrix that a validation team can execute.

Negative test	Injected condition	Expected broker/system behavior	Evidence captured
Malformed JSON	Upstream returns truncated or invalid JSON	Broker returns a structured error; code handles it	Error ledger entry, no silent coercion
Prompt injection in a database field	A free-text annotation contains instructions	Field is quarantined or escaped; no tool call outside the allow-list	Raw and sanitized payload hashes
Duplicate call	Same call arrives twice after a retry	Idempotency key returns the cached result	One upstream request, two ledger rows
Stale schema	Tool returns an older schema version	Broker rejects with a version-mismatch error	Schema version in ledger
Partial timeout	One of N parallel calls exceeds the window	Timeout error reaches the code; run is flagged incomplete	Count of N expected vs received
Oversized result	Result exceeds row or byte cap	Broker truncates with an explicit warning and a count	Record count and checksum
Expired container	Container is reclaimed mid-run	Run restarts or fails cleanly; no partial output is reported as final	Container ID chain
Forbidden data class	Result carries a direct identifier	Broker blocks and logs without forwarding	Block event, no payload retained

Each row defines a pass condition that a script can assert, and results belong in the validation package with the tool-version string and date, since timeouts and idle reclaim are described by the vendor as current values rather than guarantees.

Qualification: A/B Protocol, Evidence Pack, and Decision Matrix

A/B test protocol against direct tool calling

Anthropic itself advises teams to measure billed input tokens with and without `allowed_callers` before broad rollout. A lab-specific protocol extends that advice to the dimensions that matter in science:

1. **Select workflows:** three to five representative, non-sensitive workflows covering strong-fit, borderline, and weak-fit cases from the scorecard.
2. **Freeze inputs:** use fixed input sets and pin upstream database releases or snapshots, so differences come from the execution topology, not drifting data.
3. **Run both arms:** direct (`allowed_callers` set to direct) and programmatic (code execution only), same model, same prompts, same broker. Randomize order and run enough repeats to estimate variance.

4. **Capture the worksheet** in *Table 3* for every run.
5. **Analyze paired differences:** compare per-task paired differences, not just population averages. A published evaluation-statistics paper recommends using the paired version of the standard error estimate for model comparisons (arxiv.org).
6. **Pre-register thresholds:** define acceptance criteria before running, including a non-inferiority margin for accuracy and a floor of 100 percent for provenance completeness.

Table 3 below is the A/B worksheet. Percentage change is computed as (programmatic minus direct) divided by direct, times 100, from measured runs only.

Metric	Direct arm (measured)	Programmatic arm (measured)	% change	Note
Input tokens	record	record	(P minus D) / D x 100	Programmatic tool results are excluded from token counts
Output tokens	record	record	compute	Code generation adds output tokens
Code-execution charge	n/a	record	n/a	\$0.05 per container-hour beyond free hours
Tool charges	record	record	compute	Upstream API fees, if any
Elapsed time	record	record	compute	Fewer model round trips versus script overhead
Accuracy	record	record	compute	Against a frozen gold standard
Tool failures	record	record	compute	Timeouts, retries, schema rejects
Provenance completeness	record	record	compute	Share of calls with full contract fields
Human-review time	record	record	compute	Minutes to verify output

Token savings alone do not justify adoption: a gain that costs provenance completeness fails the protocol, and the elapsed-time and review-time columns capture costs a token meter hides. Anthropic publishes no pharma-specific results, so a lab's own paired numbers are the only valid evidence for its own tool mix.

Validation and evidence pack

The evidence pack should contain: an **intended-use statement** that names the workflow and data classes; a **requirements trace** from controls to tests; **test scripts and results**, including *Table 2* outputs; **container and version capture** (code-tool version string, model ID, SDK version, container ID chain); the **tool-call ledger** keyed on the caller field; a **deviation log**; and **change control** that re-runs the A/B subset when the model, tool version, or broker changes.

Regulatory scope should be stated precisely rather than assumed. The FDA's CSA guidance concerns computers and automated data processing systems used as part of medical device production or the quality system (www.fda.gov) and describes a risk-based approach to establish confidence in the automation

(www.fda.gov). It is directly applicable only within that scope.

For other settings, the main regulators and standards are:

- **US, Part 11:** 21 CFR Part 11 sets criteria under which electronic records are treated as trustworthy and equivalent to paper (www.ecfr.gov), requires validation of systems (www.ecfr.gov), and calls for secure, time-stamped audit trails (www.ecfr.gov); FDA has said it will narrowly interpret Part 11's scope (www.fda.gov) and recommends a justified, documented risk assessment (www.fda.gov).
- **EU:** EU GMP Annex 11 applies to all forms of computerised systems used in GMP-regulated activities (health.ec.europa.eu) and ties validation extent to risk (health.ec.europa.eu).
- **ICH:** ICH E6(R3) says computerised systems used in clinical trials should be fit for purpose (database.ich.org), with validation based on risk assessment (database.ich.org).

The MHRA material is a broader reference for evidence design. It describes core elements of a compliant data governance system across GxP sectors (www.gov.uk), defines ALCOA (attributable, legible, contemporaneous, original, accurate) (assets.publishing.service.gov.uk), states that without metadata context data has no meaning (assets.publishing.service.gov.uk), expects that users cannot amend or switch off the audit trail (assets.publishing.service.gov.uk), and expects data review to include a risk-based review of metadata (assets.publishing.service.gov.uk). For cloud services it points to understanding the service provided, ownership, retrieval, retention, and security of data (assets.publishing.service.gov.uk). PIC/S PI 041-1 cautions that validation alone does not guarantee records are adequately protected (picscheme.org), while stating that the guide is not mandatory or enforceable under law (picscheme.org). Whether any applies to a given workflow is for the organization's quality unit to determine.

Decision matrix

- **Enable PTC:** the workload scores high on fit, uses only public or contractually cleared data, and the A/B shows non-inferior accuracy and 100 percent provenance completeness.
- **Self-manage a sandbox:** the workload is a strong fit but cannot accept vendor-retained container artifacts. A self-managed sandbox keeps execution and retention inside the organization, at the cost of building the isolation Anthropic otherwise manages. Anthropic notes that agent-generated code needs sandboxing, resource limits, and monitoring (www.anthropic.com).
- **Use direct calls:** the workflow is sequential, small, or approval-driven, or needs strict-schema tools, which PTC does not support. Direct calls with prompt caching and, where permitted, batch pricing may be simpler; the Batch API offers a 50 percent discount on input and output tokens.
- **Reject for this workload:** the data class is barred from retained containers, ZDR is a hard requirement, or provenance completeness cannot reach the threshold.

Data Analysis and Evidence

Vendor measurements and their stated workloads

Vendor measurements, not forecasts. The figures below come from Anthropic, each tied to the workload it describes. None was measured on pharma, biotech, or CRO tool mixes.

- **38 percent:** roughly 38 percent fewer billed input tokens with no change in task accuracy, on a 75-tool project-management agent benchmark described as an internal evaluation.
- **20 to 40 percent:** typical token savings across production API requests with 10 to 49 tool definitions.
- **11 percent and 24 percent:** an average performance improvement of 11 percent with 24 percent fewer input tokens, when programmatic calling was added to basic search tools on web research benchmarks.
- **8 percent higher cost:** a sequential benchmark with one or two calls per turn cost roughly 8 percent more, and the page states plainly that sequential single-call workflows do not benefit.
- **37 percent:** the earlier engineering post's average token reduction on complex research tasks (www.anthropic.com), alongside accuracy gains on internal knowledge retrieval from 25.6% to 28.5% (www.anthropic.com) and on a GIA benchmark from 46.5% to 51.2% (www.anthropic.com). The post does not give sample sizes.

Illustrative arithmetic (Hypothetical Example)

The following numbers are invented to demonstrate the worksheet math. They are not measurements and must not be read as expected savings. Assume illustrative prices of \$2 per million input tokens and \$10 per million output tokens; current list prices must be taken from Anthropic's pricing page. In the direct arm, a batch-check task consumes 400,000 input tokens and 8,000 output tokens: $0.4 \times \$2 + 0.008 \times \$10 = \$0.88$. In the programmatic arm, assume 250,000 input tokens and 12,000 output tokens (more output because Claude writes code): $0.25 \times \$2 + 0.012 \times \$10 = \$0.62$, plus a 5-minute minimum container charge of $1/12 \text{ hour} \times \$0.05$, about \$0.004, for a total of about \$0.624. The percentage change is $(0.624 \text{ minus } 0.88) / 0.88 \times 100$, about negative 29.1 percent. Input tokens fall 37.5 percent while output tokens rise 50 percent. If the free 1,550 hours were not exhausted, the container charge would be zero. Other models differ: Anthropic lists Claude Opus 5.5 at \$4 and \$20 per million tokens (platform.claude.com). US-only inference adds a 1.1x multiplier (platform.claude.com), which a regional-control requirement would change.

What the research literature adds

Independent research supports the code-as-action premise but not a specific savings figure. The CodeAct paper reports up to 20 percent higher success rate than widely used alternatives (arxiv.org) with up to 30 percent fewer actions (arxiv.org), arguing that code supports control and data flow and the storage of intermediate results (arxiv.org). Those results come from general-purpose benchmarks, not from bioinformatics.

In bioinformatics, the evidence on agents is mixed and benchmark-dependent. GeneGPT, which prompts a model to call NCBI web APIs, reached an average score of 0.83 on its evaluation tasks versus 0.44 for the new Bing (arxiv.org), on a benchmark of 12 tasks with 50 question-answer pairs each (arxiv.org). BixBench's abstract (arXiv v3, revised 8 October 2025) reports that even the latest frontier models only achieve 17 percent accuracy in the open-answer regime (arxiv.org). A perturbation study found agents correctly identified corrupted inputs in 7 of 10 tasks (arxiv.org), and decoy files were used erroneously in 2 of 10 tasks (arxiv.org). These small-sample results (10 tasks each) are anecdotal, but they reinforce why the negative-test matrix includes malformed and misleading inputs. Function-calling benchmarks echo the point: memory, dynamic decision-making, and long-horizon reasoning remain open challenges (proceedings.mlr.press).

Case Studies and Real-World Examples

Published bioinformatics agent implementations

Several published systems illustrate the design space, none of them a PTC deployment. Biomni's E1 environment includes 150 specialized biomedical tools, 105 software packages, and 59 databases (www.ncbi.nlm.nih.gov). Its authors contrast code-based execution with traditional function calling, arguing the former supports the dynamic nature of biomedical workflows (www.ncbi.nlm.nih.gov), and report matching expert human performance on a database question-answering subset at 74.4 percent against 74.7 percent for experts (www.ncbi.nlm.nih.gov). BioinfoMCP converted bioinformatics tools into MCP servers, with 94.7 percent successfully executing complex workflows across three agent platforms (arxiv.org). These support the premise that code-driven orchestration is viable for scientific tools. They do not establish PTC-specific efficiency, retention posture, or validation status.

Anthropic's life-science tool surface (Neutral Overview)

Anthropic's Bio Research plugin lists instrument-data conversion supporting 40+ instrument types (claude.com), and the 10x Genomics connector is described as enabling single-cell and spatial analysis in natural language (www.anthropic.com). The announcement also positions Claude Code for processing and analyzing genomic data (www.anthropic.com), and a support article describes a library of 600+ vetted scientific tools via ToolUniverse (support.claude.com). For PTC, the practical point is that this surface is largely MCP-based, and MCP connector tools cannot be called programmatically. An Anthropic SDK helper converts MCP tools to regular API tools (platform.claude.com), but the documentation reviewed does not state how such converted tools interact with `allowed_callers`, so that route needs its own test. The container's preinstalled stack lists pandas, numpy, scipy, scikit-learn, and statsmodels, the documentation reviewed did not name bioinformatics packages, and packages cannot be installed at runtime, so alignment-scale analysis belongs in client-side pipelines.

Implications and Future Directions

For platform engineers, the main implication is that the broker becomes the system of record. Because the container is network-isolated and the model is not a trust boundary, the validated component is the client-side layer: allow-lists, schema checks, rate limits, ledger, and credentials. That layer can be qualified once and reused across direct and programmatic arms, which makes the A/B comparison fair and the evidence pack reusable.

For privacy and security leads, the 30-day retention of container artifacts is the item to escalate early. The retention table also notes that even under ZDR, flagged content may be retained for up to 2 years (platform.claude.com), so retention analysis should cover both ordinary and exceptional paths. Release notes show retention terms vary by model and feature (platform.claude.com), so the combination of model and feature must be checked, not either alone. Regional deployment choices change both cost and processing location (platform.claude.com). Platform availability matters for this retention analysis. Programmatic tool calling is listed as generally available on the [Claude API and Claude Platform on AWS, and on Microsoft Foundry for Hosted on Anthropic deployments](#). [Claude in Amazon Bedrock does not offer programmatic tool](#)

calling or server-side code execution, so Bedrock regional endpoint pricing does not apply to the programmatic arm. Anthropic's Bedrock documentation states that data handling for that offering is governed by Amazon Bedrock, so confirm retention terms with AWS for any Bedrock route and re-check retention separately for each platform before running the test.

For FinOps teams, cost has three layers (tokens, container hours, review labor), and cache hits on Opus 5.5 and Sonnet 5.5 are priced at 0.05x the base input price, so a direct-call arm with caching may narrow the gap a naive token comparison suggests. For quality teams, the practical forward path is change control tied to vendor-controlled variables. The tool version string, the idle-reclaim period described as "currently" about 5 minutes, and model retention terms can change. Re-running a small regression subset at each change is cheaper than re-validating a workflow after drift. Incorrect AI-driven predictions can mislead experimental designs (www.ncbi.nlm.nih.gov), which is why human review time belongs in the worksheet. From an advisory standpoint, the sound posture is incremental: qualify one low-risk, public-data, high-fan-out workflow first, publish the evidence internally, and expand only when the numbers justify it.

Frequently Asked Questions (FAQs)

What is Claude programmatic tool calling? Claude writes Python that calls your tools as functions inside a code execution container, and intermediate results are not loaded into the model's context.

What does `allowed_callers` do? It declares whether a tool is called directly, from code execution, or both. It is guidance, not a hard block, so the broker must enforce authorization.

Is programmatic tool calling covered by zero data retention? Not as documented in October 2026: it is built on code execution containers with data retained up to 30 days.

Does it save money? It depends on the workload. Anthropic reports about 38 percent fewer billed input tokens on one benchmark and about 8 percent higher cost on a sequential one. Tool results from programmatic calls are not counted as tokens, but code execution is billed at \$0.05 per container-hour beyond the free allotment.

Can it hold genomic data? That depends on consent, contracts, and law. Genetic data is a GDPR special category (gdpr-info.eu), and NIH controlled-access data carries NIST SP 800-171 expectations (www.grants.nih.gov). Treat genomic data as barred from the container until the data owner approves.

How should tool calls be validated? Use the tool-result contract, the negative-test matrix, and a ledger keyed on the caller field.

Is FDA CSA the governing rule? Only within its stated scope of medical-device production and quality system software (www.fda.gov). MHRA guidance is a broader reference (www.gov.uk).

Conclusion

Programmatic tool calling is best understood as a change of execution topology, not an incremental feature. The model writes the orchestration, a sandbox runs it, the client answers every tool call, and only the final output returns to context. For high-fan-out scientific workflows on public or contractually cleared data, that

topology can reduce tokens and round trips. For sequential, small, interactive, or ZDR-dependent workflows, the documented evidence points the other way.

The decisive constraint for life-science teams is retention. The feature is not ZDR eligible, and container data is retained for up to 30 days, so the data-class gate comes before any efficiency analysis. The vendor's 38 percent, 20 to 40 percent, 11 percent, and 24 percent figures belong in a hypothesis column, each tied to its stated workload; the lab's own paired measurements belong in the decision column.

A defensible rollout therefore has five parts: a fit scorecard, a data-class gate, a versioned tool-result contract, a negative-test matrix, and an A/B worksheet whose percentage changes are computed from measured runs only. Regulatory references shape how that evidence is organized, with scope confirmed by the organization's quality unit. Teams that treat the client-side broker as the validated component, and the container as a retained, untrusted compute surface, can adopt the capability without weakening provenance or controls.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. <https://platform.claude.com/docs/en/agents-and-tools/tool-use/programmatic-tool-calling>
2. <https://platform.claude.com/docs/en/release-notes/overview>
3. <https://platform.claude.com/docs/en/manage-claude/api-and-data-retention>
4. <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/computer-software-assurance-production-and-quality-management-system-software>
5. <https://www.gov.uk/government/publications/guidance-on-gxp-data-integrity>
6. <https://www.anthropic.com/news/claude-for-life-sciences>
7. <https://claude.com/plugins/bio-research>
8. <https://github.com/anthropics/knowledge-work-plugins/tree/main/bio-research>
9. <https://platform.claude.com/docs/en/agents-and-tools/tool-use/code-execution-tool>
10. <https://genai.owasp.org/llmrisk/llm062025-excessive-agency>
11. <https://www.ncbi.nlm.nih.gov/books/NBK25497>
12. <https://support.claude.com/en/articles/12614768-getting-started-with-claude-for-life-sciences>
13. <https://www.anthropic.com/engineering/advanced-tool-use>
14. <https://www.anthropic.com/engineering/code-execution-with-mcp>
15. https://www.edpb.europa.eu/system/files/2025-01/edpb_guidelines_202501_pseudonymisation_en.pdf
16. <https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/data-sharing/anonymisation/pseudonymisation>
17. <https://www.hhs.gov/hipaa/for-professionals/special-topics/de-identification/index.html>
18. <https://gdpr-info.eu/art-9-gdpr>
19. <https://gdpr-info.eu/art-4-gdpr>
20. <https://www.grants.nih.gov/sites/default/files/flmngn/NIH-Security-BPs-for-Users-of-Controlled-Access-Data.pdf>
21. <https://grants.nih.gov/policy-and-compliance/policy-topics/sharing-policies/accessing-data/requirements>
22. <https://grants.nih.gov/grants/guide/notice-files/NOT-OD-25-081.html>

23. https://ftp.uniprot.org/pub/databases/uniprot/current_release/relnotes.txt
24. <https://www.ebi.ac.uk/about/news/updates-from-data-resources/ensembl-116-release>
25. <https://www.nature.com/articles/sdata201618>
26. <https://arxiv.org/abs/2312.07852>
27. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11386446>
28. <https://www.nextflow.io>
29. <https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.100-2e2025.pdf>
30. <https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf>
31. <https://genai.owasp.org/llmrisk/llm01-prompt-injection>
32. <https://genai.owasp.org/llmrisk/llm052025-improper-output-handling>
33. https://assets.publishing.service.gov.uk/media/5aa2b9ede5274a3e391e37f3/MHRA_GxP_data_integrity_guide_March_edited_Final.pdf
34. <https://arxiv.org/pdf/2411.00640>
35. <https://www.ecfr.gov/current/title-21/chapter-I/subchapter-A/part-11>
36. <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/part-11-electronic-records-electronic-signatures-scope-and-application>
37. https://health.ec.europa.eu/system/files/2016-11/annex11_01-2011_en_0.pdf
38. https://database.ich.org/sites/default/files/ICH_E6%28R3%29_Step4_FinalGuideline_2025_0106.pdf
39. <https://picscheme.org/docview/4234>
40. <https://platform.claude.com/docs/en/about-claude/pricing>
41. <https://arxiv.org/abs/2402.01030>
42. <https://arxiv.org/html/2402.01030v4>
43. <https://arxiv.org/abs/2304.09667>
44. <https://arxiv.org/html/2304.09667>
45. <https://arxiv.org/abs/2503.00096>
46. <https://arxiv.org/html/2601.21800v2>
47. <https://proceedings.mlr.press/v267/patil25a.html>
48. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC12157518>
49. <https://arxiv.org/abs/2510.02139>
50. <https://platform.claude.com/docs/en/agents-and-tools/mcp-connector>
51. <https://platform.claude.com/docs/en/build-with-claude/claude-in-amazon-bedrock>
52. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC12476841>

THE PUBLISHER

About IntuitionLabs

IntuitionLabs is an AI consulting, custom software development and data engineering firm serving pharmaceutical, biotechnology, medical-device and other life-science organizations. We work with clinical, regulatory, medical-affairs, commercial, quality and IT teams to connect technology decisions with the work people need to accomplish.

AI consulting and adoption

Our AI enablement services cover readiness assessments, use-case selection, governance and policies, team workshops, adoption measurement and ongoing advisory support. We help organizations structure the information layer behind AI: source material, context, permissions and maintained knowledge that make generated answers useful and reviewable. Private LLM inference and hosted AI options support teams evaluating how to operate AI with appropriate control over their data and infrastructure.

Software, data and life-science workflows

IntuitionLabs develops custom software for pharma and biotech, integrates enterprise systems, and builds data engineering and business intelligence solutions. Areas of focus include AI agents, regulatory research, medical writing, medical affairs, CMC information, competitive intelligence and clinical-document workflows. Our eTMF intelligence work includes cross-system reconciliation and inspection-readiness support.

Enterprise platforms and regulated delivery

We provide Veeva services, application support, managed services, integrations and custom applications, alongside enterprise content work involving platforms such as Egnyte. For regulated workflows, our services include GxP enablement, computer-system validation and software development addressing 21 CFR Part 11 requirements. The applicable controls, validation responsibilities and acceptance criteria are defined for each engagement.

Work with IntuitionLabs

Explore [AI enablement](#), [pharma and biotech software development](#), [data engineering and BI](#), and [Veeva services](#). [Contact IntuitionLabs](#) to discuss your workflow, information sources and implementation needs.

IntuitionLabs publishes educational research to help life-science teams make informed technology decisions. Coverage of a product or organization does not imply a client relationship, endorsement or partnership.

Website: <https://intuitionlabs.ai>

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.