IntuitionLabs

INTUITIONLABS / RESEARCH & PRACTICAL GUIDANCE

Claude Bioinformatics Agents: VirBench Viral Retrieval

> 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 · 2026-11-02 · claude bioinformatics agents · virbench benchmark · gget virus · ncbi virus api · viral sequence retrieval · deterministic retrieval · llm agents bioinformatics · claude code bioinformatics · data provenance

Original article: https://intuitionlabs.ai/articles/claude-bioinformatics-agents-virbench-viral-sequence-retrieval



In this article

1. Executive Summary
2. Introduction and Background
3. Key Changes: From Agent-Built Datasets to Agent-Orchestrated Retrieval
4. Implementation Considerations and Process Changes
5. Data Analysis and Evidence
6. Case Studies and Real-World Examples
7. Implications and Future Directions
8. Frequently Asked Questions (FAQs)
9. Conclusion

Executive Summary

The practical question behind "Claude bioinformatics agents" is whether a large language model (LLM) agent may assemble a scientific sequence dataset on its own, or whether it should only orchestrate a deterministic, versioned retrieval tool. The most direct public evidence is **VirBench**, a benchmark of **120 manually curated viral sequence queries** described in an arXiv preprint submitted on **4 June 2026** (arxiv.org). Without a dedicated retrieval layer, mean accuracy ranged from **16.9%** for Claude Sonnet 4 to **91.3%** for GPT-5.5 (arxiv.org). Claude Opus 4.7 reached **83.2%** (arxiv.org). With the deterministic tool **gget virus**, Opus 4.7 reached **98.3%** (arxiv.org).

The failure modes are specific and testable: under-counting from incomplete pagination, over-counting from misapplied filters, and sharp degradation as filters multiply (arxiv.org). Reference counts came from manual runs in the NCBI Virus web interface (github.com), and each query carries a fixed maximum release date to keep expected counts stable (github.com). The consequences travel downstream: in Anthropic's published example, an incomplete retrieved set pushed an inferred common-ancestor date back to 1922 (www.anthropic.com).

The architectural lesson is that filtering, pagination, batching and logging belong in versioned code. gget virus applies sequential server-side and local filters (pachterlab.github.io) and writes an execution summary with statistics, output files and errors (pachterlab.github.io). The benchmark pinned gget version 0.30.3 (github.com); PyPI lists 0.30.8 as the current release as of this report. Treat 0.30.3 as the benchmark's pinned version, not a recommendation: qualify and pin whichever version you adopt, and rerun a canary query set whenever it changes (pypi.org).

This report translates those findings into a query contract, a dataset manifest, negative tests, a rerun policy and a build, adopt or reject matrix. Provenance practice is anchored in the FAIR principles, which ask that data carry detailed provenance (www.nature.com), and in NIST guidance that accountability presupposes transparency (nvlpubs.nist.gov). It also states limits: one viral benchmark, scored on counts, run on specific model versions, authored with disclosed support from a model provider. The results are a snapshot and should not be generalized to all of bioinformatics. Nothing here is outbreak or treatment guidance.

Introduction and Background

Dataset construction is usually treated as plumbing. For pathogen-genomics, vaccine and platform teams it is a scientific control point: the sequences chosen determine every tree, alignment, conservation map and training set computed afterward. When an LLM agent performs that step, the control point moves into a system whose output can differ between runs of the same prompt.

Anthropic has positioned Claude for life sciences work through connectors, Agent Skills and Claude Code, naming genomic data processing with Claude Code as a use case (www.anthropic.com). Claude Science, a separate public beta app rather than a model, is described as able to query more than 60 scientific databases (claude.com). Capability is therefore not the open question. The open question is which steps an agent should perform free-form and which it should delegate to deterministic code.

The VirBench work addresses one such step, retrieving viral sequences from the National Center for Biotechnology Information (NCBI). The preprint defines deterministic retrieval as invariance to pagination, batching or response ordering. NCBI Virus is not a database itself but a portal aggregating multiple underlying resources, which is why a filter that is one click in a web interface can become a multi-API script for an agent. NCBI describes the portal as a community resource for viral sequence data from RefSeq, GenBank and other NCBI repositories (www.ncbi.nlm.nih.gov).

This report is written for platform leads, computational biologists, research software engineers, data stewards and AI governance partners. As an adjacent advisor rather than a vendor of retrieval software, IntuitionLabs frames the decision as a data-governance question: what contract must a dataset satisfy before an analysis may rely on it, and which component is allowed to certify that contract? The sections below follow the standard-analysis pattern: key changes, implementation considerations, quantitative evidence, neutral examples, and implications. This report reproduces no benchmark ground-truth values.

Key Changes: From Agent-Built Datasets to Agent-Orchestrated Retrieval

Change 1: A benchmark makes retrieval measurable

VirBench evaluates how well AI agents and tools retrieve viral sequence data (github.com). Each agent receives a natural-language query with filters and must return the number of matching sequences (github.com). The design choices matter more than the headline:

- **Breadth:** the queries span 40 pathogens across multiple taxonomic levels (arxiv.org).
- **Filter load:** each query applies between 1 and 9 simultaneous filters (median 6) (arxiv.org).
- **Fixed release dates:** every query except three accession lookups carries a maximum release date, intended to increase the stability of expected counts over time (github.com).
- **Repeated runs:** each prompt was repeated three times (arxiv.org).
- **Manual ground truth:** the maintainers state that every reference value comes from running the query by hand in the NCBI Virus web interface (github.com).

Two curation details are useful for any team building its own acceptance set. The protocol notes that the web interface moves the start date by one day after submission (github.com), and that where the web interface and the tool disagree, the primary GenBank record is authoritative (github.com). Reference values are never edited in place (github.com); nine of the 120 differ between curation versions 1 and 2 (github.com), and each arm is scored against the version current when it ran (github.com).

Change 2: Agent-only failure modes are named, not anecdotal

The preprint attributes under-counting to incomplete pagination and premature termination, and notes that for some agents over-counting was more frequent than under-counting, which it links to filter-application errors (arxiv.org). Agents defaulting to the API must retrieve data in batches of up to 1,000 records (arxiv.org).

Anthropic's account, which comes from a party that also funded the preprint and so is not an independent replication, reports the same pattern from the user's point of view: repeated identical questions produced substantially different answers (www.anthropic.com). This is a statement about a specific configuration, an agent using web search and sandboxed Python whose default outbound allowlist permits only NCBI and PyPI domains (github.com). It should not be read as a general verdict on any model.

A related, more general concern is identifier fidelity. Practitioner guidance for curators observes that LLMs frequently produce identifiers in valid formats that point to nothing (ai4curation.github.io), because the models are good at following format patterns (ai4curation.github.io). No peer-reviewed measurement of invented viral accessions was found in this research, so that point is stated as a design precaution, not a measured rate. The VirBench failures are of a different kind: the retrieved records are real, but the set is incomplete or wrongly filtered. A format check on accessions would not detect that.

Change 3: gget virus moves filtering, pagination and logging into versioned code

The documentation describes gget virus as applying sequential server-side and local filters to download customized datasets (pachterlab.github.io). Its architecture, as reported in the preprint, combines three NCBI interfaces:

- **Metadata** through the NCBI Datasets REST API.
- **Feature annotations** through GenBank records retrieved via E-utilities, with GenBank metadata retrieval enabled automatically when a filter needs it (pachterlab.github.io).
- **Large taxa** such as SARS-CoV-2 and Influenza A through NCBI's optimized cached data packages (pachterlab.github.io), using the NCBI datasets command-line binary that ships bundled with gget (pachterlab.github.io).

When a taxon does not support a filter server-side, the tool removes the unsupported filter and reapplies it locally (arxiv.org). Outputs are FASTA, CSV and JSON Lines files in an output folder (pachterlab.github.io), and every run writes an execution summary of statistics, output files and errors (pachterlab.github.io).

On NCBI collaboration wording: Anthropic states that gget virus was developed in collaboration with researchers at NCBI (www.anthropic.com). The preprint's own language is narrower, and the gget documentation page itself does not state an endorsement. Teams should therefore describe the tool as a community-maintained client of NCBI interfaces, not as an NCBI product.

Change 4: Model choice matters much less once retrieval is deterministic

Anthropic summarizes the finding as follows: adding a deterministic retrieval layer made model choice much less important (www.anthropic.com). The numbers support this, as the Data Analysis section shows. Anthropic's engineering guidance names wrong tool selection and incorrect parameters as the most common tool-use failures (www.anthropic.com), which matches the benchmark's qualification that a tool must also be called correctly. Most residual errors came from agent post-processing, and some agents ignored the instruction and reverted to other APIs in a small number of runs (detailed in the validation protocol below).

The same pattern appears in the wider literature on tool-augmented models. GeneGPT, which let a model call NCBI E-utilities, scored 0.83 on its benchmark against 0.44 for a retrieval-augmented search chatbot (www.ncbi.nlm.nih.gov), and its authors concluded that database utility tools might be superior to relevant web pages (www.ncbi.nlm.nih.gov). ChemCrow reported a similar benefit from expert-designed tools in chemistry, while cautioning that response validity depends on the quality and quantity of the tools (www.nature.com).

Implementation Considerations and Process Changes

A reproducible query contract

A query contract is the artifact that turns a request into something testable. It should be written before any retrieval, signed off by the data steward, and passed unchanged to the deterministic tool. The fields below mirror the filter vocabulary the gget documentation exposes: host by name or taxonomy identifier (pachterlab.github.io), nucleotide completeness (pachterlab.github.io), ambiguous-character ceilings (pachterlab.github.io), length bounds (pachterlab.github.io), collection dates (pachterlab.github.io), a maximum release date (pachterlab.github.io), geography (pachterlab.github.io), source database (pachterlab.github.io), and inclusion or exclusion of lab-passaged (pachterlab.github.io) and vaccine-strain sequences (pachterlab.github.io).

```
{
  "contract_version": "1.0",
  "purpose": "assay-coverage reference set (illustrative)",
  "taxon": {"name": "<taxon name>", "taxid": "<NCBI taxonomy id>"},
  "accessions": null,
  "source_database": "genbank",
  "max_release_date": "YYYY-MM-DD",
  "collection_date": {"min": "YYYY-MM-DD", "max": "YYYY-MM-DD"},
  "geography": "<value, with the field-matching rule noted>",
  "host": "<name or taxid>",
  "nuc_completeness": "complete",
  "max_ambiguous_characters": 0,
  "length": {"min": 0, "max": 0},
  "exclusions": {"lab_passaged": true, "vaccine_strain": true},
  "tool": {"name": "gget virus", "version": "0.30.3", "mode": "pinned"},
  "expected_artifacts": ["sequences.fasta", "metadata.csv", "command_summary.txt", "manifest.json"],
  "independent_check": {"method": "manual NCBI Virus run", "sample": "<queries or accessions>"}
}
```

Several semantics need explicit decisions in the contract, because the underlying interfaces are not uniform:

- **Date boundaries:** the NCBI Virus collection-date filter requires both a from and a to date, and the web interface shifts a start date after submission (github.com). State the inclusive or exclusive rule and test it.
- **Geography:** the gget geographic filter emulates NCBI Virus by searching across location, region and virusName fields (arxiv.org). A country query can therefore match differently from a field-exact query.
- **Completeness and host:** completeness is determined by the submitter (www.ncbi.nlm.nih.gov), and the host filter is the submitter-provided host organism, not the known host range of a virus (www.ncbi.nlm.nih.gov). Neither is a curated truth claim.
- **RefSeq and GenBank:** the two record types are mutually exclusive in NCBI Virus (www.ncbi.nlm.nih.gov). RefSeq is a curated, non-redundant collection (www.ncbi.nlm.nih.gov), so mixing it with GenBank changes redundancy.
- **Accession identity:** an accession without a version suffix always refers to the latest version of the sequence (www.ncbi.nlm.nih.gov), and the version increments when the sequence changes. Contracts should store accession.version, not bare accessions.

Dataset manifest

The manifest records what was actually produced. It should be generated by the tool wrapper, not by the agent. FAIR principles aim at enhancing the ability of machines to automatically find and use the data (www.nature.com), and ask for globally unique, persistent identifiers (www.nature.com), detailed provenance (www.nature.com) and a clear usage license (www.nature.com); the manifest is where those requirements become fields. For metadata vocabulary, the Genomic Standards Consortium's MIxS standard is for reporting contextual metadata about sequenced samples (genomicsstandardsconsortium.github.io) and includes a checklist for Minimum Information About an Uncultivated Virus Genome (genomicsstandardsconsortium.github.io).

```
{
  "manifest_version": "1.0",
  "contract_sha256": "<hash of the signed contract>",
  "retrieved_at_utc": "<timestamp>",
  "database_cutoff": {"max_release_date": "YYYY-MM-DD"},
  "tool": {"gget": "0.30.3", "datasets_cli": "<version>"},
  "agent": {"model_id": "<id>", "run_id": "<id>", "role": "orchestrator only"},
  "filters_echo": {"...": "copied from the tool summary, not retyped"},
  "record_count": 0,
  "accession_list": {"file": "accessions.txt", "sha256": "<hash>", "format": "accession.version, sorted"},
  "sequences": {"file": "sequences.fasta", "sha256": "<hash>"},
  "metadata": {"file": "metadata.csv", "sha256": "<hash>"},
  "exclusions": [{"rule": "lab_passaged", "count": 0}],
  "tool_log": {"file": "command_summary.txt", "sha256": "<hash>", "errors": 0, "retries": 0},
  "independent_check": {"result": "pass|fail", "method": "<text>"}
}
```

The hash function is a deliberate choice. NIST's Secure Hash Standard specifies SHA-256 among its algorithms (nvlpubs.nist.gov), while the European Nucleotide Archive documents that any change to a file's contents changes its MD5 checksum (ena-docs.readthedocs.io). Either supports fixity; SHA-256 is the more conservative default for new internal manifests. For packaging, the RO-Crate community describes a lightweight approach to packaging research data with metadata (www.researchobject.org), and a Workflow Run profile exists to capture the provenance of the execution of computational workflows (www.researchobject.org). The World Wide Web Consortium's PROV overview explains that provenance information can be used to form assessments about quality, reliability or trustworthiness (www.w3.org). The gget execution summary already records statistics, output files and errors (pachterlab.github.io), so the wrapper's job is to hash it and link it, not to recreate it.

Validation protocol and negative tests

The deterministic tool lowers risk; it does not remove the need to test it, because the most common residual failure mode in the benchmark was incorrect local filtering after download (arxiv.org). A practical protocol combines positive checks with negative tests that are designed to fail:

1. **Independent reconciliation:** for a sampled subset of contracts, compare the tool's count and accession set with a manual NCBI Virus run, mirroring the benchmark's own ground-truth method.
2. **Pagination stress:** include at least one contract whose result exceeds the default page sizes. ESearch returns only 20 identifiers by default and caps retmax at 10,000 records (www.ncbi.nlm.nih.gov); the benchmark notes API batches of up to 1,000 records.
3. **Monotonicity:** adding a filter must never increase the count. This is a metamorphic test that needs no ground truth.
4. **Partition additivity:** because GenBank and RefSeq records are mutually exclusive (www.ncbi.nlm.nih.gov), their counts should sum to the combined count. Date partitions should add up once records with missing dates are counted separately.
5. **Repeat-run identity:** run the same contract three times with the same release cutoff and require byte-identical sorted accession lists, not merely equal counts.
6. **Silent-fallback test:** expose only the approved tool to the agent and fail the run if any other retrieval route is used. The documented reversion to other APIs in a small number of runs shows why prompting alone is insufficient (arxiv.org).
7. **Upstream outage test:** simulate an NCBI-side error and require a hard failure with logged retry information. The gget documentation shows a sample message for temporary NCBI server-side issues (pachterlab.github.io).
8. **Post-processing audit:** reject any agent-side filtering after download unless it is itself a logged, tested tool step, since incorrect local filtering was the leading residual failure.
9. **Canary queries:** keep a small sealed set of contracts with known counts for regression runs, and keep their answers out of any prompt, document or repository the agent can read. The VirBench maintainers withhold their reference values for the same reason.

Provenance, checksums, logging and rerun policy

Reproducibility guidance for computational research is consistent on the basics: keep track of how each result was produced (pmc.ncbi.nlm.nih.gov) and archive the exact versions of all external programs used. For AI-enabled systems, NIST's AI Risk Management Framework states that trustworthy AI depends upon accountability and that accountability presupposes transparency (nvlpubs.nist.gov). The generative AI profile suggests inventory entries that include data provenance information (nvlpubs.nist.gov) and documenting how content provenance data is tracked (nvlpubs.nist.gov). These are voluntary frameworks, not mandates, and teams should consult their own quality and compliance functions on which controls apply.

A rerun policy should state triggers, not a calendar:

- **Tool change:** rerun the canary and a reconciliation sample whenever the pinned gget or NCBI Datasets version changes.
- **Cutoff change:** moving the maximum release date creates a new dataset, not a refresh. GenBank issues a full release every two months (www.ncbi.nlm.nih.gov), but the international partners exchange data daily (www.ncbi.nlm.nih.gov), so "latest" is a moving target.
- **Incremental updates:** gget's baseline option downloads only accessions not found in the baseline (pachterlab.github.io). Store the diff (added, removed, version-changed) as its own artifact.
- **Interface change:** rerun negative tests when NCBI documentation or rate-limit policies change.

Operations: rate limits and large downloads

Retrieval layers must respect NCBI's usage rules. E-utilities guidance is no more than three URL requests per second without a key (www.ncbi.nlm.nih.gov) and up to 10 per second by default with an API key (www.ncbi.nlm.nih.gov). The Datasets API and command-line tools default to 5 requests per second (www.ncbi.nlm.nih.gov), rising to 10 with a key. NCBI's scripting guidance asks for larger jobs on weekends or between 9 pm and 5 am Eastern Time on weekdays (www.ncbi.nlm.nih.gov). A wrapper can enforce these limits centrally, whereas per-request scripts must each re-implement them. Batch metadata requests in gget default to 200 records (pachterlab.github.io).

On licensing, NCBI states that it places no restrictions on the use or distribution of the data (www.ncbi.nlm.nih.gov), while noting that submitters may assert their own rights. The benchmark data are gated and recipients agree not to republish the reference values (github.com).

Data Analysis and Evidence

Results by agent, model and access route

Table 1 below consolidates the agents, their access routes and the paper's accuracy figures. Accuracy is defined per query as the fraction of the three runs that exactly match the expected sequence count (arxiv.org), then averaged across queries. Stability is a per-query bounded coefficient of variation computed across runs (arxiv.org). Baseline figures are from the paper's results text (arxiv.org), gget virus figures likewise (arxiv.org), and the final column is our own arithmetic on the reported means.

System (as reported)	Access route and tools	Accuracy without gget virus	Accuracy with gget virus	Points gained (computed)
Claude Sonnet 4	Anthropic Messages API; web search and Python execution; used because later models carried biosafety-related access restrictions on viral sequence retrieval (github.com)	**16.9%**	**92.8%**	+75.9
Claude Opus 4.7	Anthropic Messages API; queries of this class were restricted by default pending a provider-side access review completed in May 2026 (github.com)	**83.2%**	**98.3%**	+15.1
Biomni v0.0.8	A data-lake agent powered by Claude Sonnet 4; free text converted to counts by a secondary parsing step	**22.5%**	**90.0%**	+67.5
Edison Analysis	A proprietary agent accessed via a client SDK	**40.0%**	**93.1%**	+53.1
GPT-5.2-pro	OpenAI model evaluated in the same benchmark	**67.1%**	**98.9%**	+31.8
GPT-5.5	OpenAI model evaluated in the same benchmark	**91.3%**	**99.7%**	+8.4

Table 1 should be read with four cautions. First, **Claude Sonnet 4 and Claude Opus 4.7 are different results** and should never be merged into "Claude accuracy": the gap between them without the tool is 66.3 points. Second, the arms ran at different times against reference versions that differ in some values (nine of 120 between curation versions 1 and 2, noted above), so cross-row comparisons carry that uncertainty. Third, the paper presents 95% confidence intervals graphically, and the text reviewed for this report gives no numeric intervals, so none are reproduced here; teams that need them should recompute from per-run results after requesting the gated deposit (github.com). Fourth, run retention differs slightly by arm, and runs with explicit errors are excluded from metrics (github.com).

What the arithmetic shows

Three derived observations follow from the reported means. The spread between the weakest and strongest system was 74.4 points without the tool (91.3 minus 16.9) and 9.7 points with it (99.7 minus 90.0). The residual miss rate, defined here as 100 minus accuracy, fell from 16.8 to 1.7 points for Opus 4.7 (about 9.9-fold) and from 83.1 to 7.2 points for Sonnet 4 (about 11.5-fold). For GPT-5.5 it fell from 8.7 to 0.3 points (about 29-fold). Because each query is worth 1/120 of the mean (0.83 points) and each run 1/360 (0.28 points), differences below roughly one query should not be over-interpreted.

Stability tells a complementary story. With the tool, stability across systems ranged from 0.92 to 1.00 (arxiv.org). Stability and accuracy are separate properties, as the worked example below demonstrates.

Repeated-run variance and exact-match calculations

Table 2 below is a **(Hypothetical Example)** with invented counts, not benchmark data. Assume a query whose expected count is 100, and use the population standard deviation (the paper's exact convention should be checked before reuse). Stability is $1/(1+\sigma/\mu)$.

Three-run outcome	Exact-match accuracy	Mean / σ	Stability	Interpretation
100, 100, 100	1.00	100 / 0.00	1.000	Correct and repeatable
90, 100, 110	0.33	100 / 8.16	0.925	Unbiased mean, but only one exact match
60, 100, 100	0.67	86.7 / 18.86	0.821	Two exact matches; one under-count
80, 80, 80	0.00	80 / 0.00	1.000	Perfectly stable and consistently wrong

The last row is the governance trap: a stable pipeline can be stably incomplete. Stability should therefore be reported only alongside exact-match accuracy against an independent reference. Count equality is also necessary but not sufficient for dataset identity, because two different accession sets can share a count. For that reason the acceptance worksheet later in this report compares sorted accession.version sets, not only totals.

Volume and logging evidence

The data-transfer claims are specific. In one example, SARS-CoV-2 sequences released in 2025 containing the surface glycoprotein required downloading 284 GB before local filtering (arxiv.org), against 3.8 GB through gget virus (arxiv.org). The arithmetic: (284 minus 3.8) divided by 284 is 98.7%, consistent with the more than 98% reduction the authors report for high-volume queries. The authors also ran a stress test of all 120 queries, three at a time concurrently every hour for 24 hours (Figure 4). It compared programmatic retrieval methods (the NCBI Datasets REST API, the Datasets CLI and gget virus), not LLM agents, and is a useful template for a reader's own repeat-run test of a retrieval layer.

Evidence quality notes

- **Preprint status:** the arXiv listing shows a single version, and this report did not find a peer-reviewed version.
- **Disclosed interests:** the paper states that it uses Claude models and was partially funded by Anthropic through its AI for Science program. It also lists an Anthropic-affiliated co-author, reports Anthropic compute credits, and thanks both Anthropic and OpenAI for support (arxiv.org).
- **Withheld benchmark:** the full dataset is not publicly released, to prevent leakage into model training corpora (github.com); it is available to researchers on request (github.com), so independent reproduction needs the gated deposit.
- **Prompt transparency:** the system prompt requires exactly one integer on its own line (github.com), and the manuscript reports Claude runs with web search enabled (github.com).

Case Studies and Real-World Examples

Phylogenetic timing: why an incomplete set matters

Anthropic's account compares trees built from a manually curated NCBI Virus set and from sets an agent retrieved in three identical runs. Two of the three agent-retrieved sets were visibly incomplete, and one tree pushed the inferred time to the most recent common ancestor (TMRCA) back to 1922 (www.anthropic.com).

This report does not reproduce any benchmark ground-truth counts. The general mechanism is well established in the phylogenetics literature: taxon sampling affects divergence-time estimates (pmc.ncbi.nlm.nih.gov), and in a study of rabbit hemorrhagic disease virus, using even one misdated taxon in a large dataset could significantly skew estimates (pmc.ncbi.nlm.nih.gov).

Test for your own pipeline: rebuild the tree or date estimate from the three repeated retrievals and from the reference set, and record the spread of the estimated root date. If the spread exceeds the precision your decision needs, the retrieval step has not met the contract. Nothing about this test requires a clinical or outbreak claim.

Assay coverage and target variation

The same dependency exists for target-variation summaries and for diagnostic and assay design. An in silico reassessment of published SARS-CoV-2 assays found primer or probe mismatches in 7 assays, a result that depends on which sequences are used as the reference set (pmc.ncbi.nlm.nih.gov). In bacterial work, primer coverage was overestimated when computed against a reference database biased by PCR amplification (www.ncbi.nlm.nih.gov). These examples are not about agents; they show that coverage is a function of the sequence set, so the set must be contract-defined and reproducible. A practical test is to compute coverage on the three repeated retrievals and on the reference set, and to compare the number of unmatched sequences.

Training sets and leakage

Sequence-based machine-learning sets add a separate hazard. Standard random splitting can give dubious assessments of generalization because of similarity between samples (arxiv.org). A retrieval step that is incomplete or unstable changes which near-duplicates enter each split. The contract should therefore record redundancy handling, and the manifest should freeze the accession list before any clustering or splitting, so that train and test partitions can be regenerated exactly.

Implications and Future Directions

Architecture comparison and decision matrix

Table 3 compares three architectures for the retrieval step, and gives a build, adopt or reject reading for each. The assessments are analytic judgments built on the cited evidence, not measured results for any specific product.

Architecture	Completeness and filter semantics	Repeatability and provenance	Operational fit	Reading
A. Agent browses or calls APIs directly	Depends on the agent finishing pagination and applying filters; under-counting and over-counting observed in the benchmark	Repeated runs of one prompt can differ (see Change 2); no guaranteed manifest	Fast to prototype; unsuitable as the sole source for regulated or publication-bound datasets	**Reject** for dataset construction; acceptable for exploration with labeled output

Architecture	Completeness and filter semantics	Repeatability and provenance	Operational fit	Reading
B. Agent writes general API glue per request	Possible but depends on each script; multi-batch downloads can stop partway, so the script must handle resumption itself	Scripts can be saved and hashed, but each request may produce a different script	Must manage rate limits and batches in every script (www.ncbi.nlm.nih.gov)	**Build only** with a reviewed, versioned library; otherwise treat as Architecture A
C. Agent orchestrates a deterministic, versioned tool	Filters resolved locally where servers do not support them; Opus 4.7 reached 98.3% with the tool (Table 1)	Pinned version, execution summary, logged errors (pachterlab.github.io)	Depends on external NCBI infrastructure; needs wrapper tests	**Adopt** with contract, manifest and negative tests

Table 3 shows why the right unit of decision is the retrieval step, not the model. Architecture C still shows residual error (for example 98.3% for Opus 4.7, not 100%), so the contract and independent checks stay in place. There is also no public threshold that says how much residual error a given downstream use can tolerate, and this report does not invent one.

Acceptance worksheet for reader-run evaluation

Instead of a universal pass mark, teams should set a threshold per downstream use and record the rationale. A minimal worksheet:

- **Use case and tolerance:** state what decision the dataset feeds and what count or composition error it can absorb.
- **Sample:** pick the number of contracts, covering small, large (above page limits) and multi-filter queries.
- **Reference method:** name the independent route and who executed it.
- **Metrics:** exact-match accuracy per contract over at least three runs, stability, accession-set identity, and runtime.
- **Decision rule:** written before the run, with any waiver and approver recorded.
- **Re-baseline trigger:** model, tool, cutoff or interface changes.

When the pattern generalizes, and when it does not

The pattern transfers most directly to databases that share the traits that made the viral case hard: filter logic available only in a web interface, paginated or capped API responses, submitter-provided metadata, and versioned records. The NCBI interfaces show several such caps, including a 10,000-record ceiling per ESearch request (www.ncbi.nlm.nih.gov).

It does not transfer automatically to tasks that are not retrieval. BioAgent Bench, an evaluation of end-to-end bioinformatics tasks such as RNA-seq, variant calling and metagenomics, found that correct high-level pipeline construction does not guarantee reliable step-level reasoning (arxiv.org). BixBench reported frontier models achieving only 17% accuracy in an open-answer regime on analytical scenarios (arxiv.org), and BioML-bench found that biomedical specialization did not confer a consistent advantage (www.biorxiv.org). Those are different tasks with different metrics, so they inform caution without being comparable to VirBench.

Other caveats apply. VirBench scores counts, so it cannot certify sequence content. It tests specific models in 2026 and the authors themselves describe the results as a snapshot (arxiv.org). In one run (Q32, run 3), GPT-5.5 found and used gget virus without being prompted to; a single observation does not show a trend, and a contract is still needed for repeatable retrieval. Finally, access policy is part of the environment: the benchmark team used Sonnet 4 because newer models carried biosafety-related access restrictions (github.com), so governance teams should confirm current permissions with the model provider rather than assume the benchmark configuration is available.

Claude-specific implementation paths

For teams using Claude, the benchmark points to a division of labor. Anthropic describes Programmatic Tool Calling as letting Claude orchestrate tools through code (www.anthropic.com), with loops, conditionals and error handling explicit in code rather than implicit in the model's reasoning (www.anthropic.com). Agent Skills are folders of instructions, scripts and resources (www.anthropic.com), and Anthropic notes that bundling code suits steps where traditional programming is more reliable than token generation (www.anthropic.com). Skills are intended to let Claude consistently follow specific protocols (www.anthropic.com), and Anthropic maintains a plugin marketplace of life-sciences skills (github.com). Tool use itself works by Claude determining when to call a tool based on the user's request (docs.claude.com), and the Model Context Protocol is an open-source standard for connecting AI applications to external systems (modelcontextprotocol.io). Anthropic also reports that wrong tool selection and incorrect parameters are the most common tool-use failures (www.anthropic.com), which is consistent with the benchmark's residual errors.

A Claude Code workflow that follows from this: wrap the pinned gget call in a script, expose only that script to the agent, have the wrapper emit the manifest, and have the agent explain, orchestrate and downstream-analyze. Anthropic's guide separates Claude Code, used when the output is software that ships to other teams, from Claude Science, used for analyses (claude.com); Claude Science is described as carrying an auditable history of how each output was made (www.anthropic.com) and as including a reviewer agent that checks citations and calculations (www.anthropic.com). Product-level provenance is helpful, but it records what happened; it does not decide what a dataset was supposed to contain. That remains the contract's job.

The commonly sought query "claude bioinformatics agents" is therefore best answered by a pattern, not a product: deterministic retrieval, explicit contract, manifest, negative tests, and an agent that explains and orchestrates but does not certify the dataset.

Frequently Asked Questions (FAQs)

What is the VirBench benchmark?

VirBench is a benchmark for evaluating how well AI agents and tools retrieve viral sequence data (github.com), with 120 manually curated queries across 40 pathogens. Agents return a count of matching sequences (github.com), compared with manually curated expected counts from the NCBI Virus web interface (github.com).

Can Claude retrieve viral sequences reliably on its own?

In the benchmark configuration, Claude Opus 4.7 reached 83.2% mean accuracy without a retrieval layer and 98.3% with gget virus; Claude Sonnet 4 and the other systems are in Table 1. Those are results for specific models and a counting task, not a general rating.

Why use deterministic sequence retrieval with LLM agents?

Because filter logic, pagination and logging can be tested and versioned in code. The benchmark found sharp degradation beyond three to four simultaneous filters for agent-only retrieval.

How do LLM agents for bioinformatics perform more broadly?

Results vary by task. BioAgent Bench found that correct pipeline construction does not guarantee reliable step-level reasoning, and BioML-bench found that biomedical specialization did not confer a consistent advantage (www.biorxiv.org).

What does Claude for life sciences offer here?

Anthropic describes genomic data processing with Claude Code (www.anthropic.com) and skills for following protocols consistently (www.anthropic.com). These are orchestration capabilities; the benchmark suggests pairing them with a deterministic retrieval tool.

How does an agent retrieve sequences from NCBI Virus?

NCBI Virus is a portal over multiple resources. Programmatic access combines the Datasets REST API, command-line packages and E-utilities. Datasets genome downloads can be limited to genomes released on or after a date (www.ncbi.nlm.nih.gov).

How can teams guard against invented or wrong accessions?

Verify that every identifier exists by querying an authoritative API (ai4curation.github.io), pin accession.version (www.ncbi.nlm.nih.gov), and check set completeness, which an existence check alone does not cover.

Does tool use help in other sciences?

ChemCrow's authors report that expert-designed tools can mitigate model hallucination in chemistry (www.nature.com). Chemistry is a different task and metric from sequence retrieval.

Conclusion

The VirBench and gget virus work gives dataset teams a concrete, evidence-based reason to separate two roles that are easy to conflate. An LLM agent is useful for planning, explaining, orchestrating and analyzing. A deterministic, versioned tool is better suited to certifying what a database query returned. In the benchmark, the retrieval layer lifted every tested system to at least 90.0% mean accuracy and narrowed the spread between systems from 74.4 to 9.7 points.

The decision for a platform lead is therefore not "which model," but "which component is allowed to define the dataset." A defensible answer has five parts: a signed query contract, a pinned deterministic tool, a machine-generated manifest with checksums and logs, negative tests that are designed to fail, and a rerun policy tied to tool, cutoff and interface changes. Residual errors remain, and the evidence is limited to one viral benchmark scored on counts, on specific model versions, in a preprint with disclosed provider support. Teams should treat the results as a strong prompt for local testing, not a transferable guarantee.

Finally, scope matters. This report concerns data-engineering controls for sequence retrieval. It offers no outbreak, clinical or treatment guidance, and organizations with biosafety, quality or compliance obligations should involve those functions when adopting any agentic workflow that touches pathogen data.

Source links

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

1. https://arxiv.org/abs/2606.06749
2. https://arxiv.org/html/2606.06749
3. https://github.com/lauraluebbert/VirBench/blob/main/docs/curation_protocol.md
4. https://github.com/lauraluebbert/VirBench
5. https://www.anthropic.com/research/agents-in-biology
6. https://pachterlab.github.io/gget/en/virus.html
7. https://github.com/lauraluebbert/VirBench/blob/main/requirements.txt
8. https://pypi.org/project/gget
9. https://www.nature.com/articles/sdata201618
10. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf
11. https://www.anthropic.com/news/claude-for-life-sciences
12. https://claude.com/science
13. https://www.ncbi.nlm.nih.gov/labs/virus
14. https://ai4curation.github.io/aidocs/how-tos/make-ids-hallucination-resistant
15. https://www.anthropic.com/engineering/advanced-tool-use
16. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10153281
17. https://www.nature.com/articles/s42256-024-00832-8
18. https://www.ncbi.nlm.nih.gov/labs/virus/vssi
19. https://www.ncbi.nlm.nih.gov/refseq/about
20. https://www.ncbi.nlm.nih.gov/genbank/sequenceids
21. https://genomicsstandardsconsortium.github.io/mixs
22. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
23. https://ena-docs.readthedocs.io/en/latest/submit/fileprep/preparation.html
24. https://www.researchobject.org/ro-crate
25. https://www.researchobject.org/workflow-run-crate

26. https://www.w3.org/TR/prov-overview
27. https://www.ncbi.nlm.nih.gov/books/NBK25499
28. https://pmc.ncbi.nlm.nih.gov/articles/PMC3812051
29. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf
30. https://www.ncbi.nlm.nih.gov/genbank
31. https://www.ncbi.nlm.nih.gov/genbank/collab
32. https://www.ncbi.nlm.nih.gov/books/NBK25497
33. https://www.ncbi.nlm.nih.gov/datasets/docs/v2/api/api-keys
34. https://www.ncbi.nlm.nih.gov/home/about/policies
35. https://github.com/lauraluebbert/VirBench/blob/main/docs/prompts.md
36. https://pmc.ncbi.nlm.nih.gov/articles/PMC3370833
37. https://pmc.ncbi.nlm.nih.gov/articles/PMC3426481
38. https://pmc.ncbi.nlm.nih.gov/articles/PMC7353963
39. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3445835
40. https://arxiv.org/pdf/2402.14482
41. https://arxiv.org/abs/2601.21800
42. https://arxiv.org/abs/2503.00096
43. https://www.biorxiv.org/content/10.1101/2025.09.01.673319v1.full
44. https://www.anthropic.com/news/skills
45. https://github.com/anthropics/life-sciences
46. https://docs.claude.com/en/docs/agents-and-tools/tool-use/overview
47. https://modelcontextprotocol.io/introduction
48. https://claude.com/resources/guides/claude-science-product-guide/when-to-use-which-claude-surface
49. https://www.anthropic.com/news/claude-science-ai-workbench
50. https://www.ncbi.nlm.nih.gov/datasets/docs/v2/reference-docs/command-line/datasets/download/virus/genome

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.