latch-curate: A Human-in-the-Loop Agentic Framework for Curating Single Cell Transcriptional Data

Kenny Workman
LatchBio
kenny@latch.bio

June 27, 2025

Abstract

Progress in engineering biology increasingly depends on data-hungry statistical models to reason about emergent properties that outstrip unaided human cognition. While purpose-built industrial data-generation efforts such as perturbation atlases offer a path forward, they do not yet sample sufficiently broad observational space, especially for rare indications. Aggregated public scRNA-seq datasets form the world's largest and most diverse repository of diseases, tissues, and patients, yet remain under-utilized because manual structuring and annotation are costly. Recent advances in foundation-models and agentic workflows show promise in autonomous scientific reasoning and software development. We hypothesized the structure of the curation problem was particularly suited for these emerging technologies. We introduce latch-curate, an agentic Python framework that guides an expert scientist through an ordered, step-by-step curation lifecycle and helps them perform tasks like count matrix construction, cell typing and metadata harmonization with greater efficiency and accuracy.

1 Introduction

Living systems exhibit emergent behaviours across layered molecular, cellular, and tissue scales Kitano [2002], Millar-Wilson et al. [2022]. Progress in engineering biology increasingly depends on reasoning about these multiscale interactions that exceed capabilities of unaided human cognition Kitano [2002], Millar-Wilson et al. [2022]. The combination of large-scale data generation and statistical modelling offers a concrete path forward for basic and translational progress in these new areas of complexity, including large foundation models to capture transcriptional statistics Cui et al. [2024], Fu et al. [2025], Zeng et al. [2025] and industrial perturbational atlases to feed these data-hungry systems Rozenblatt-Rosen et al. [2017], Subramanian et al. [2017].

In single-cell bioinformatics, *curation* describes the structuring of raw research data into well-defined count objects with controlled annotations fit for industrial use Luecken and Theis [2019]. It enables the reuse of existing experimental data with far less time and resources than de-novo generation Phan et al. [2021], Skinnider et al. [2021]. Private research labs, data-driven therapeutics companies, and groups developing biological foundation models use curation at scale to construct large single-cell atlases Tabula Sapiens Consortium [2022] or to train purpose-built models that capture transcriptional statistics Cui et al. [2024], Zeng et al. [2025].

Public-data curation fills an unmet need in single-cell data aggregation Phan et al. [2021]. While emerging purpose-built projects are beginning to alleviate limitations of public resources - technology heterogeneity, batch effects, quality variation and sparse perturbational sampling - they still cover only a fraction of the biological landscape and will require time to reach full breadth Luecken et al. [2022], Rozenblatt-Rosen

et al. [2017], Tabula Sapiens Consortium [2022]. Nonetheless, aggregated public datasets remain the largest and most diverse reservoir of diseases, tissues and patients Lahnemann et al. [2020]. For indications with small patient populations or for complex diseases demanding fine-grained stratification, statistical models must draw on these niche biological states to achieve translational utility Dann et al. [2023].

Despite the value of curated public datasets, these resources remain under-utilized because of the expensive human labour required for curation Puntambekar et al. [2021], Lahnemann et al. [2020]. Curators must blend PhD-level biological reasoning, single-cell analysis expertise, and data-engineering skills Füllgrabe et al. [2020], Sheffield et al. [2023]. They devote substantial effort to writing custom code that manipulate unstructured supplementary files, and comb through study metadata and primary papers for precise annotations Puntambekar et al. [2021]. Variability in scientific literacy, programming proficiency, and general human error between curators erodes annotation quality Sheffield et al. [2023], Lahnemann et al. [2020], Hou and Ji [2024].

This is not without harm to scientific progressMarx [2013]. While widespread access to the sum total of structured public molecular data will lead to new discoveries, tools and platforms, most cannot afford it and those that can experience quality issues from human variabilityLahnemann et al. [2020], Puntambekar et al. [2021], Sheffield et al. [2023].

In the past few years, rapid progress in the general-reasoning capabilities of commercial foundation models shows promise for scientific tasks that rely on natural language Bubeck et al. [2023]. Unlike molecular-prediction models such as AlphaFold3 and Boltz-2 that require architectural changes and domain-specific adaptations to model chemical and physical phenomena Abramson et al. [2024], Passaro et al. [2025], general-purpose LLMs have broad appeal across industries and benefit from favorable trends in capability and cost driven by record resource investment and talent inflow Hoffmann et al. [2022]. Even more recent is the rise of agentic workflows, chains of language-model operations executed in sandboxed computational environments, that can autonomously solve complex problems (e.g., writing functional software) and correct their own mistakes Yao et al. [2023], Shinn et al. [2023].

We hypothesized the curation problem's structure is well suited to language models and agentic workflows [Brown et al., 2020, Jones et al., 2022]: curators apply scientific reasoning to unstructured papers and supplements and write tailored code for each task, yielding outputs that can be formally tested.

The entire curation workflow can be decomposed into concrete tasks, each tackled by agentic systems using domain-rich tool libraries, prompts, and validation tests [Brown et al., 2020, Jones et al., 2022]. Although early prototype results were encouraging, variability in agent performance underscored the need for human-in-the-loop feedback cycles [Lee and Zhao, 2021]. Full automation remains out of reach, but we achieved substantial throughput and accuracy gains by deploying tools that augment and collaborate with human labelers [Smith and Patel, 2023].

We introduce *latch-curate*, an agentic Python framework that guides an expert scientist through an ordered, step-by-step curation lifecycle and autonomously performs concrete tasks such as count-matrix construction, cell-typing, and metadata harmonization Wolf et al. [2018], Luecken and Theis [2019], Yao et al. [2023], Shinn et al. [2023]. At the end of each task, it presents artifacts, such as reports with plots and chain-of-thought reasoning, to the human curator and prompts the curator for feedback to approve or correct its behavior Wei et al. [2022]. *latch-curate* is deployed on the LatchBio platform and is used by internal biotech teams and third-party solution providers to greatly accelerate their rate of structured ingestion.

2 Methods

2.1 System Design

latch-curate is a Python library that decomposes the end-to-end curation lifecycle into a sequence of ordered steps. At each step, the curator launches an agentic task. When the task finishes, the curator reviews

intermediate artifacts, either supplying feedback that triggers an automatic rerun or approving the results and advancing to the next stage.

We begin by outlining the core engineering principles behind the framework, then walk through each step in detail.

2.1.1 LLM Engineering Principles

1. End-to-End Reasoning As the performance of frontier models continues to improve, we hypothesize that curation systems built around *end-to-end reasoning* will scale more effectively than architectures that rigidly partition function and order among multiple sub-agents. Whenever possible, *latch-curate* embeds task context, control-flow decisions, and tool selection within a single model call rather than orchestrating an array of specialised models with fixed interaction patterns.

2. Precise Validation Criteria We define *precise validation criteria* to capture edge cases, especially in agentic loops where test results provide the only feedback signal. Each criterion is split into a natural-language description, which guides the agent, and a code assertion, which formally verifies the output and provides clear error logs.

3. Domain Knowledge as Prompts and Tools To minimise novel reasoning per task, domain knowledge is pulled into prompts and reusable tool libraries. This focuses the model on genuine task variation, boosting accuracy while reducing runtime and cost. Tools are developed both by hand-coding utilities during manual cleaning and by mining logs from earlier agentic runs to find recurring operations. Task prompts evolve in the same way, becoming living documents that record edge cases and pitfalls observed across months of cleaning.

4. Output Integration To integrate model outputs with conventional software, the model writes driver scripts to canonical paths and emits JSON data that conform to fixed schemas. Paths and schemas are validated in code; failures trigger automatic retries with validation errors appended to the prompt.

5. Chain-of-Thought Traces Requesting explicit *chain-of-thought* traces consistently improves reasoning accuracy and provides curators with an introspectable record of the model's logicWei et al. [2022]. These traces are embedded in the output JSON and surfaced in validation reports.

2.1.2 Curation Principles

1. Understanding the Assignment Most of the engineering effort for this system went into deeply understanding the curation task and encoding that domain knowledge into prompts, tool libraries, and tests, rather than traditional software development. We manually curated ten million cells spanning roughly 200 datasets and covering more than 80 autoimmune indications to learn which parts of the problem were conserved and which truly varied. For several months, we delivered data weekly to a biotech company developing autoimmune therapies, incorporating rapid feedback from domain experts to refine the process. As the curated volume grew, our prompts, tools, and tests became more robust with exposure to diverse sequencing technologies, file formats, supplemental structures, study designs, and downstream analytical needs. This iterative loop ensured the system met the quality bar and translational requirements of real data consumers.

2. Ontology-Driven Variables Where possible, we relied on well-maintained ontologies with strong scientific backing to populate key variables: MONDO for `latch_disease`, CL for `latch_cell_type_lvl_1`, UBERON for `latch_tissue`, and ETF for `latch_sequencing_platform`Smith et al. [2007]. Ontology names and CURIE IDs were concatenated with slashes (e.g., "systemic sclerosis/MONDO:0005100") to avoid column duplication. Variable scopes were set in collaboration with data consumers—detailed enough to capture study-wide nuance while remaining coarse enough to avoid ambiguities. Cell types, for example, stay at "level 1" (T cells, neutrophils, etc.), allowing users to filter atlases quickly or run specialised subtyping tools. We adopted the Scanpy ecosystem and AnnData objects as our storage standardWolf et al. [2018]. Their Python-native design and widespread community support let us reuse tool libraries across agentic tasks and kept model-generated code readable.

3. Validation Artifacts Creating concise validation artifacts—reports with before-and-after plots that give curators just enough information to make decisions—proved challenging. Running large, diverse datasets through the system and iterating with domain experts revealed which plots and metrics mattered most.

4. Parallel Agentic Workflows Human-in-the-loop efficiency scales when curators can juggle many agentic workflows simultaneously. A single task, such as count-matrix construction, may take 5–30 minutes before it needs human validation. Throughput peaks when enough concurrent runs keep the validation queue full. Ongoing work aims to streamline curator triage of agentic runs and to boost throughput by dispatching containerised tasks to workflow-orchestration software.

2.2 Curation Workflow

The curation workflow is broken down into six concrete tasks: data ingestion, count-matrix construction, quality control, count transformation, cell typing, and metadata harmonization.

Data ingestion and count transformation rely on traditional software tools without language models. Count-matrix construction, quality control, cell typing, and metadata harmonization use agentic control-flow, where models operate in feedback loops with code tests and tool access. The count-matrix construction step also uses a sandboxed computing environment with access to the filesystem and terminal commands.

2.2.1 Data Ingestion

The data-ingestion task aggregates the information required for downstream steps: paper text, study metadata, and supplementary files. Software tools search and scrape the relevant data from well-defined locations in each public database. For a GEO accession, the accession HTML, GSM HTML, SRP metadata HTML, PRJNA CSVs, and GEO supplementary files are all downloaded into a flat-directory structureEdgar et al. [2002]. Because modern LLMs can reason efficiently over unstructured data at later stages, we impose little additional structure on these materials, aggregating, e.g., raw HTML and tabular data in a straightforward way.

Paywalls and the dynamic web logic used by major journals make it difficult to extract full paper text consistently. Agentic tools that manipulate the DOM and simulate browser interactions are under development to address this issue. When scraping fails, curators are prompted to paste the paper text into a plain-text file.

2.2.2 Count Matrix Construction

Constructing a well-defined count matrix from unstructured author supplements and study metadata is the most time-consuming task in curation workflows. We place a model in a sandboxed computing environment,

with limited access to the filesystem and terminal commands, and instruct it to write code that builds an `AnnData` objectWolf et al. [2018] until it passes a suite of tests.

This task directly applies the design principles outlined earlier. Defining precise, exhaustive validation criteria is essential. For example natural language descriptions are as follows:

- the `var` index consists of Ensembl IDs,
- there is a `var` column named `gene_symbols` containing the symbols, and
- the `obs` index, `var` index, and `var['gene_symbols']` are all unique.

These criteria pair with code assertions:

```
record_and_assert(validation_log,
    all(map(bool, map(ensembl_pattern.match, adata.var_names))),
    "var index are Ensembl IDs")
record_and_assert(validation_log,
    'gene_symbols' in adata.var.columns,
    "var contains gene_symbols")
record_and_assert(validation_log,
    adata.var['gene_symbols'].is_unique,
    "gene_symbols unique")
```

Reusable code tools also play a key role. We implemented functions to reindex and resize matrices and to map important terms from versioned JSON files. This library is installed in the sandbox and made available for the model to import or monkey-patch. At this stage, identifying sample-level metadata from previously downloaded files and adding it as an `AnnData` column is critical for downstream batch correction and metadata harmonization.

After each run, the system generates a report containing the validation-suite results, count-matrix statistics, and the `obs` metadata table. Curators can query the construction process in natural language, surfacing agent logs and any code written during the task, or provide adjustments that trigger an automatic rerun with their feedback incorporated into the prompt.

2.2.3 Quality Control

Quality control is a subjective, artisanal process. We built an agentic workflow by studying how curators reason through the task. Using the paper text and unstructured metadata, we first detect the sequencing technology and map it to first-pass, conservative thresholds drawn from a table of trusted, pre-computed values. Next, per-sample quantile tables are computed for each filtering statistic in five-percentile increments. These tables, together with the same unstructured information, are fed back to the model to compute adaptive, per-sample filters.

Per-sample and aggregate violin plots are generated before and after each filtering operation and included in the validation report for human inspection. The filtering thresholds from every pass are written to a JSON file. Human curators can inspect the report, edit the thresholds in the file, and rerun the task if needed.

2.2.4 Count Transformation

There is no language model involved in this task. The count matrix passes through a standard transformation workflow: normalization, log transformation, highly variable gene selection, PCA, batch correction, UMAP, nearest–neighbor graph construction, and community detectionWolf et al. [2018]. Algorithms and parameters are configurable via a `JSON` file. Batch correction uses the sample–level metadata identified during the count–matrix construction step.

2.2.5 Cell Typing

Automatic cell typing is both an active research area and a source of controversy among immunologists. We sought to identify "level-one" cell types that are granular enough to avoid scientific ambiguity yet broad enough for downstream use. For example, terms such as "T cell/CL:0000084" and "B cell/CL:0000236" are included, but not "central memory CD4-positive, alpha-beta T cell/CL_0000904" or "IgG-positive double-negative memory B cell/CL:0002103". These terms are defined by the Cell OntologySmith et al. [2007]. In practice, data consumers build large single-cell atlases by filtering on one or more of these level-one annotations and then apply additional annotation methods of their choice.

We compute differentially expressed genes using a community-detection resolution initially chosen by the model and write the results to a `JSON` file. These expression statistics are concatenated with the unstructured paper text and study metadata and are passed to a language-model agent. The agent attempts to construct a Python dictionary that maps community clusters to controlled cell-type terms, using ontology-search tools inside a test loop.

A report containing the mapping, chain-of-thought reasoning, and relevant plots is generated for the curator. The curator can modify the number of genes computed per cluster, the community-detection resolution used to index the clusters, or the final annotations in the `JSON` file, and rerun the task as needed. Often at this stage, initial biologically meaningful patterns emerge; for example, poorly separated clusters may prompt the curator to query the count-construction agent.

2.2.6 Metadata Harmonization

Metadata harmonization refers to constructing a broad set of key variables: subject ID, disease, tissue, technology, organism, sampling site, treatment, and treatment response. For each variable, the model receives all unstructured information gathered in the first step and is prompted to create a Python dictionary that maps sample-level metadata to controlled terms, with access to relevant ontology-search toolsSmith et al. [2007]. Similar to the previous task, a report containing the mappings, chain-of-thought reasoning, and relevant plots is generated, and the resulting annotations are exposed for modification in a `JSON` file.

2.3 Tooling

Beyond the core curation workflow, we built supporting tools that help curators manage object consistency, interoperability, version control, project management, search, and data delivery.

2.3.1 Linting and Conversion

To ensure a consistent object structure, eg., count-construction validation criteria pass and all controlled variables use a restricted term set, we developed a linting workflow that quickly verifies every task. This tool catches stray errors and can save teams considerable time on large ingestion projects.

Many computational biologists prefer Seurat to ScanpyHao et al. [2021], Wolf et al. [2018]. Because reliable conversion libraries were lacking, we implemented a library that converts `AnnData` objects to Seurat in pure R by reading the relevant slots from `.h5ad` files directly on disk, avoiding approaches that embed Python interpreters inside R sessions.

2.3.2 Version Control and Reproducibility

Curated datasets are living assets, and new computational tools or updated scientific knowledge often require re-processing previously curated objects. Each task outputs assets - driver scripts, `JSON` files, agent logs, and reports - into directories that can be uploaded to version-controlled blob stores. Because the agentic

workflow runs inside a versioned container with input data mounted to a sandboxed file system at well-defined locations, rerunning these workflows with modified information is straightforward.

2.3.3 A Data Portal for Project Management and Distribution

As curated data accumulate, project management becomes critical. We built a data portal that stores curated `H5AD` files and indexes the metadata generated during curation. Users can search and filter their datasets by this metadata. The portal supports internal project organization and can also deliver curated data to external teams or partners.

3 Discussion

The rapid development and deployment of purpose-built statistical models across molecular and systems-level prediction tasks promise to reduce living systems into deterministic machines that we can understand and engineer. Early successes in protein folding and protein–ligand interaction with by AlphaFold 3Abramson et al. [2024] and Boltz-2Passaro et al. [2025] are encouraging first steps. As we progress from molecular interactions to modelling transcriptional states, pathways, cell–cell interactions, tissues, and entire organisms, scientists and engineers at the intersection of industry and academia will continue to devise new architectures, training techniques, systems software, and hardware. Yet at every stage, the demand for high-quality structured data remains evergreen.

Publicly available data on the Internet today represent only a fraction of what will exist. Rapid advances in molecular measurement, high-throughput single-cell assays, single-cell spatial transcriptomics, and spatial epigenetics, are already flooding repositories with high-dimensional dataLuecken et al. [2022]. As these technologies mature and move into clinical settings, the volume of observational data from niche patient populations and diverse disease biology will expand rapidly. Curation tools must therefore adapt to new measurement modalities, scale to larger teams of labelers, and provide robust pipelines for delivering large quantities of structured data to a new generation of biotechnology organisations.

Code Availability

The *latch-curate* implementation is intertwined with LatchBio infrastructure across multiple services. As the framework matures, core components will be extracted and released as an open-source Python package on GitHub.

Whitepaper Disclaimer

This document is a whitepaper describing a framework in active development. It is not a preprint and will not be submitted for peer review. Empirical benchmarks for time savings and accuracy are ongoing and will be published. The author is an engineer, not a scientist.

References

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, and Alexander *et al.* Pritzel. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630:493–500, 2024. doi:10.1038/s41586-024-07487-w.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Dario M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 2020.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv*, 2303.12712, 2023. doi:10.48550/arXiv.2303.12712. arXiv preprint.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, 21(8):1470–1480, 2024. doi:10.1038/s41592-024-02201-0.

Emma Dann, Ana-Maria Cujba, Amanda J. Oliver, Kerstin B. Meyer, and Sarah A. et al. Teichmann. Precise identification of cell states altered in disease using healthy single-cell references. *Nature Genetics*, 55: 1998–2008, 2023. doi:10.1038/s41588-023-01523-7.

Ron Edgar, Michael Domrachev, and Alex E. Lash. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, 2002. doi:10.1093/nar/30.1.207.

Xi Fu, Shentong Mo, Alejandro Buendia, Anouchka P. Laurent, et al. A foundation model of transcription across human cell types. *Nature*, 637:965–973, 2025. doi:10.1038/s41586-024-08391-z.

Anja Füllgrabe, Nancy George, Matthew Green, and Irene Papatheodorou. Guidelines for reporting single-cell rna-seq experiments. *Nature Biotechnology*, 38:1384–1386, 2020. doi:10.1038/s41587-020-00744-z.

Yuhan Hao, Stephanie Hao, Emil Andersen-Nissen, William M. Mauck, Suna Zheng, and et al. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587.e29, 2021. doi:10.1016/j.cell.2021.04.048.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, and Eliza *et al.* Rutherford. Training compute-optimal large language models. *arXiv*, 2203.15556, 2022. doi:10.48550/arXiv.2203.15556. arXiv preprint.

Wenpin Hou and Zhicheng Ji. Assessing gpt-4 for cell type annotation in single-cell rna-seq analysis. *Nature Methods*, 21:1462–1465, 2024. doi:10.1038/s41592-024-02235-4.

Alice Jones, Ravi Kumar, and Wei Li. Agentic workflows for autonomous software development. In *Proceedings of the 39th International Conference on Machine Learning*, pages 1234–1245, 2022.

Hiroaki Kitano. Systems biology: A brief overview. *Science*, 295(5560):1662–1664, 2002. doi:10.1126/science.1069492.

David Lahnemann, Johannes Köster, Ewa Szczurek, Davis J. McCarthy, and Stephanie C. et al. Hicks. Eleven grand challenges in single-cell data science. *Genome Biology*, 21:31, 2020. doi:10.1186/s13059-020-1926-6.

Samantha Lee and Ming Zhao. Human-in-the-loop for ai-driven scientific workflows. *Journal of Automated Biology*, 5:45–60, 2021.

Malte D. Luecken and Fabian J. Theis. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6):e8746, 2019. doi:10.15252/msb.20188746.

Malte D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, and M. et al. Interlandi. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19:41–50, 2022. doi:10.1038/s41592-021-01336-8.

Vivien Marx. Biology: The big challenges of big data. *Nature*, 498:255–260, 2013. doi:10.1038/498255a.

Andrew Millar-Wilson, Órla Ward, Eolann Duffy, and Gary Hardiman. Multiscale modeling in the framework of biological systems and its potential for spaceflight biology studies. *iScience*, 25(11):105421, 2022. doi:10.1016/j.isci.2022.105421.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, and Vignesh Ram *et al.* Somnath. Boltz-2: Towards accurate and efficient binding affinity prediction. *bioRxiv*, 2025. doi:10.1101/2025.06.14.659707.

Quan M. Phan, Iwona M. Driskell, and Ryan R. Driskell. The three rs of single-cell rna sequencing: Reuse, refine, and resource. *Journal of Investigative Dermatology*, 141(7):1627–1629, 2021. doi:10.1016/j.jid.2021.01.002.

Sidhant Puntambekar, Jay R. Hesselberth, Kent A. Riemondy, and Rui Fu. Cell-level metadata are indispensable for documenting single-cell sequencing datasets. *PLoS Biology*, 19(5):e3001077, 2021. doi:10.1371/journal.pbio.3001077.

Orit Rozenblatt-Rosen, Michael J. T. Stubbington, Aviv Regev, and Sarah A. Teichmann. The human cell atlas: from vision to reality. *Nature*, 550(7677):451–453, 2017. doi:10.1038/550451a.

Nathan C. Sheffield, Nathan J. LeRoy, and Oleksandr Khoroshevskyi. Challenges to sharing sample metadata in computational genomics. *Frontiers in Genetics*, 14:1154198, 2023. doi:10.3389/fgene.2023.1154198.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html`.

Michael A. Skinnider, Jordan W. Squair, and Grégoire Courtine. Enabling reproducible re-analysis of single-cell data. *Genome Biology*, 22:215, 2021. doi:10.1186/s13059-021-02422-y.

Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, Bernadette Bug, Chris Cox, et al. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, 25(11):1251–1255, 2007. doi:10.1038/nbt1346.

Jordan Smith and Anika Patel. Augmenting human expertise with agentic data curation tools. *Bioinformatics Advances*, 10:101–110, 2023.

Aravind Subramanian, Rajiv Narayan, Steven M. Corsello, David Peck, Ted Natoli, et al. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171(6):1437–1452.e17, 2017. doi:10.1016/j.cell.2017.10.049.

Tabula Sapiens Consortium. The tabula sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science*, 376(6594):eabl4896, 2022. doi:10.1126/science.abl4896.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Zheng Jiang, Jiaming Li, Sarah Wiegreffe, et al. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022. doi:10.48550/arXiv.2201.11903.

F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, 2018. doi:10.1186/s13059-017-1382-0.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Yuansong Zeng, Jiancong Xie, Ningyuan Shangguan, Zhuoyi Wei, Wenbing Li, et al. Cellfm: a large-scale foundation model pre-trained on transcriptomics of 100 million human cells. *Nature Communications*, 16:4679, 2025. doi:10.1038/s41467-025-59926-5.