The Journal of Open Source Software

Tataki: Enhancing the robustness of bioinformatics workflows with simple, tolerant file format detection

Masaki Fukui [1], **Hirotaka Suetake** [1], **and Tazro Ohta** [2,3]

1 Sator, Inc. **2** Institute for Advanced Academic Research, Chiba University **3** Department of Artificial Intelligence Medicine, Graduate School of Medicine, Chiba University

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 25 December 2025
Published: unpublished

Summary

Tataki is a lightweight command-line tool for workflow-oriented file format detection in bioinformatics. Automated genomics pipelines frequently exchange large intermediate files (e.g., SAM/BAM/CRAM, VCF, BED), which can sometimes be empty, truncated, or structurally inconsistent. Because many downstream tools do not reliably detect such conditions, these malformed files can lead to silent failures that are hard to diagnose. Tataki inspects actual file contents using strict, domain-aware parsers to identify malformed or mixed-format files that general-purpose tools often misclassify, and includes an extension mode based on the Common Workflow Language (CWL) to support custom or emerging formats. It emphasizes verifying that each file matches the expected format at its workflow stage, aiming to improve workflow robustness and provenance.

Statement of need

Modern bioinformatics workflows integrate many specialized analytical tools to process large scale sequencing data (Perkel, 2019). These workflows are designed to reduce manual intervention and to ensure reproducibility by automating complex multi-step analyses. However, despite their widespread adoption, workflow executions often fail for surprisingly simple reasons: intermediate files may be empty, truncated, or internally inconsistent, and many bioinformatics tools do not reliably signal such failures through exit codes (Niu et al., 2022). As a consequence, downstream tools may misinterpret file formats, propagate incorrect assumptions, or terminate unexpectedly. These silent errors reduce the overall robustness and fault tolerance of automated workflows, which becomes increasingly problematic as datasets grow and manual oversight becomes infeasible.

This fragility is rooted in longstanding characteristics of the bioinformatics ecosystem. The field contains a large and heterogeneous collection of file formats, many of which lack formal specifications or have multiple variants used by different tools. Beyond a handful of well-standardized formats such as SAM/BAM (Li et al., 2009)/CRAM (Cochrane et al., 2012), VCF (Danecek et al., 2011) or BED (Niu et al., 2022), many commonly used formats have ambiguous boundaries or are interpreted differently across implementations (Rehm et al., 2021). At the same time, bioinformatics software is often developed by individual research groups for specific tasks, typically originating as stand-alone research code created under tight time and resource constraints, leading to inconsistent behaviors, uneven error handling, and non-standard assumptions about input data (Brack et al., 2022). These factors combine to create a landscape in which workflows must routinely exchange files whose correctness cannot be assumed.

To address these challenges, we developed Tataki, a lightweight and workflow-friendly file format detection tool tailored to bioinformatics. Tataki is designed to combine strict parsers

with an extensive mode for structurally variable formats, aiming to support both standarized and real-world data irregularities. By incorporating `Tataki` between workflow steps, researchers and developers can rapidly identify format-related anomalies before they propagate, and improve debugging of multi-step pipelines. Ultimately, this leads to more robust and reproducible research, and enables more accurate provenance recording for workflow outputs.

Overview of `Tataki`

`Tataki` is written in Rust and designed as a workflow component rather than an interactive validator. It examines file content directly without relying on filename extensions and handles compressed inputs in `gzip` and `bzip2` formats by decompressing them before analysis. To detect anomalous or truncated files that superficially appear valid, `Tataki` scans entire files. However, given that genomics intermediate files can reach hundreds of gigabytes, it also provides bounded inspection (e.g., checking only the first N records or lines) to reduce latency. Results are presented in a structured manner using terms from the EDAM ontology (Black et al., 2022).

`Tataki` supports two complementary detection modes. Its native mode uses strict, domain-aware parsers for major genomics formats. For greater flexibility, the External Extension Mode allows users to define format identification logic via CWL (Crusoe et al., 2022).

External Extension Mode

The External Extension Mode enables users to define custom file format identification logic through the CWL, which adds support for emerging or project-specific formats without modifying the core software. In this mode, format recognition is delegated to a user-supplied CWL document, which specifies how a file should be processed and which EDAM format identifier should be assigned upon successful validation.

External Extension Mode allows `Tataki` to remain lightweight while accommodating the diverse and evolving landscape of file formats used in bioinformatics research. This approach provides a flexible pathway for integrating domain-specific validators and facilitates more reliable workflow execution in specialized or rapidly developing research areas.

State of the field

Existing file format detection tools only partially address the challenges of robust file validation in genomics workflows. General-purpose utilities such as `file` (Darwin, n.d.), Magika (Fratantonio et al., 2025), Siegfried (Lehane, n.d.), and TrID (Pontello, n.d.) rely on static magic byte rules, trained classifiers, heuristics, or other techniques, and have minimal or no support for domain-specific bioinformatics formats. Due to these algorithmic characteristics, they often misclassify genomics files with mixed or mislabeled content, typically identifying only the first recognizable format segment.

PipeVal (Patel et al., 2024) is a tool developed specifically for use in bioinformatics workflows. It performs quick validations using format-specific modules selected based on file extensions, and includes checksum-based comparisons to detect file corruption. However, its scope differs from `Tataki`: PipeVal assumes files are correctly typed and verifies internal consistency, whereas `Tataki` focuses on independently verifying that the output format truly matches what is expected at that workflow stage, even in the presence of malformed or hybrid files.

Given this landscape, extending existing tools would not address the structural limitations in format coverage, content verification, or pipeline integration.

Software Design

Tataki was designed with an emphasis on execution flexibility and extensibility in workflow environments. Extensibility here refers both to the ability for users to introduce project-specific format checks and to keeping the codebase approachable for community contributions.

To support flexibility and extensibility, format detectors are implemented as independent modules, each responsible for a single file format. This structure allows detectors to be composed and reordered without affecting the core logic. In principle, such a design could enable dynamic loading of user-developed detectors, reducing the size of the core binary and allowing third-party extensions to be deployed without recompilation.

However, dynamically loading in Rust requires Rust's unsafe features, and introduces additional constraints on version compatibility, testing, and deployment. In heterogeneous workflow environments, these factors undermine reproducibility and complicate continuous integration and distribution.

Instead of relying on native dynamic loading, Tataki adopts External Extension Mode as its extensibility mechanism. This design balances flexibility with robustness, aligning Tataki's extensibility model with the reliability requirements of automated bioinformatics workflows.

Research Impact Statement

Tataki improves the robustness of bioinformatics workflows by detecting malformed, truncated, or structurally inconsistent intermediate files before they propagate to downstream analysis steps.

Tataki is under consideration for integration into an existing peer-reviewed workflow execution service developed within our organization. In this setting, Tataki would act as a shared preflight layer, enabling format checks and machine-readable provenance without requiring users to modify their workflow definitions. This provides benefits such as earlier failure detection, reduced waste of computational resources, and more reliable provenance records across diverse workflow engines.

Tataki is designed to be community-ready. Format detectors are implemented as independent modules, and the repository provides documentation, templates, and tests to facilitate the addition of new detectors. The project is released under the Apache-2.0 license. For ease of adoption, Tataki is distributed as a single static binary and as an OCI container image, allowing straightforward integration into local environments and container-based workflow platforms.

Limitations

Tataki focuses on identifying file formats and detecting structural anomalies; it does not perform semantic validation of biological content. For example, it does not verify whether sequence identifiers are consistent across related files or whether genomic coordinates fall within valid ranges. Such checks remain the responsibility of downstream analysis tools or dedicated validation software.

AI Usage Disclosure

Generative AI tools were used in a limited manner during the development of Tataki and the preparation of this manuscript. During software development, generative AI was used to assess the feasibility of design ideas, clarify specific implementation approaches, and generate code snippets. No agentic or autonomous AI systems were used for coding. The codebase was

primarily written, reviewed, and validated by MF, with minor contributions from other human collaborators.

Project documentation was primarily written by MF, with help from generative AI for grammar checks and clarification of wording and tone. For manuscript preparation, generative AI was used to assist with drafting text. All AI-assisted content was subsequently reviewed, edited, and verified by the authors to ensure technical accuracy and consistency with the software implementation and cited literature.

Acknowledgements

This work was supported by the research grant from the SECOM Science and Technology Foundation (FY2023 grant program).

References

Black, M., Lamothe, L., Eldakroury, H., Kierkegaard, M., Priya, A., Machinda, A., Singh Khanduja, U., Patoliya, D., Rathi, R., Che Nico, T. P., Umutesi, G., Blankenburg, C., Op, A., Chieke, P., Babatunde, O., Laurie, S., Neumann, S., Schwämmle, V., Kuzmin, I., … Kalaš, M. (2022). *EDAM: The bioscientific data analysis ontology (update 2021)*. https://doi.org/10.7490/f1000research.1118900.1

Brack, P., Crowther, P., Soiland-Reyes, S., Owen, S., Lowe, D., Williams, A. R., Groom, Q., Dillen, M., Coppens, F., Grüning, B., Eguinoa, I., Ewels, P., & Goble, C. (2022). Ten simple rules for making a software tool workflow-ready. *PLOS Computational Biology*, *18*(3), 1–11. https://doi.org/10.1371/journal.pcbi.1009823

Cochrane, G., Alako, B., Amid, C., Bower, L., Cerdeño-Tárraga, A., Cleland, I., Gibson, R., Goodgame, N., Jang, M., Kay, S., Leinonen, R., Lin, X., Lopez, R., McWilliam, H., Oisel, A., Pakseresht, N., Pallreddy, S., Park, Y., Plaister, S., … Zalunin, V. (2012). Facing growth in the European Nucleotide Archive. *Nucleic Acids Research*, *41*(D1), D30–D35. https://doi.org/10.1093/nar/gks1175

Crusoe, M. R., Abeln, S., Iosup, A., Amstutz, P., Chilton, J., Tijanić, N., Ménager, H., Soiland-Reyes, S., Gavrilović, B., Goble, C., & Community, T. C. (2022). Methods included: Standardizing computational reuse and portability with the Common Workflow Language. *Communications of the ACM*, *65*(6), 54–63. https://doi.org/10.1145/3486897

Danecek, P., Auton, A., Abecasis, G., Albers, C. A., Banks, E., DePristo, M. A., Handsaker, R. E., Lunter, G., Marth, G. T., Sherry, S. T., McVean, G., Durbin, R., & Group, 1000. G. P. A. (2011). The variant call format and VCFtools. *Bioinformatics*, *27*(15), 2156–2158. https://doi.org/10.1093/bioinformatics/btr330

Darwin, I. (n.d.). *Fine Free File Command (and libmagic) — darwinsys.com*. Retrieved February 6, 2026, from https://www.darwinsys.com/file/

Fratantonio, Y., Invernizzi, L., Farah, L., Thomas, K., Zhang, M., Albertini, A., Galilee, F., Metitieri, G., Cretin, J., Petit-Bianco, A., Tao, D., & Bursztein, E. (2025). MAGIKA: AI-Powered Content-Type Detection. *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, 2638–2649. https://doi.org/10.1109/ICSE55347.2025.00158

Lehane, R. (n.d.). *IT for archivists — itforarchivists.com*. Retrieved February 6, 2026, from https://www.itforarchivists.com/siegfried

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., & Subgroup, 1000. G. P. D. P. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, *25*(16), 2078–2079. https://doi.org/10.1093/bioinformatics/btp352

Niu, Y. N., Roberts, E. G., Denisko, D., & Hoffman, M. M. (2022). Assessing and assuring interoperability of a genomics file format. *Bioinformatics*, *38*(13), 3327–3336. https://doi.org/10.1093/bioinformatics/btac327

Patel, Y., Beshlikyan, A., Jordan, M., Kim, G., Holmes, A., Yamaguchi, T. N., & Boutros, P. C. (2024). PipeVal: Light-weight extensible tool for file validation. *Bioinformatics*, *40*(2), btae079. https://doi.org/10.1093/bioinformatics/btae079

Perkel, J. M. (2019). Workflow systems turn raw data into scientific knowledge. *Nature*, *573*(7772), 149–150. https://doi.org/10.1038/d41586-019-02619-z

Pontello, M. (n.d.). *Marco Pontello's Home - Software - TrID — mark0.net*. Retrieved February 6, 2026, from https://mark0.net/soft-trid-e.html

Rehm, H. L., Page, A. J. H., Smith, L., Adams, J. B., Alterovitz, G., Babb, L. J., Barkley, M. P., Baudis, M., Beauvais, M. J. S., Beck, T., Beckmann, J. S., Beltran, S., Bernick, D., Bernier, A., Bonfield, J. K., Boughtwood, T. F., Bourque, G., Bowers, S. R., Brookes, A. J., … Birney, E. (2021). GA4GH: International policies and standards for data sharing across genomic research and healthcare. *Cell Genomics*, *1*(2), 100029. https://doi.org/10.1016/j.xgen.2021.100029