BioHackrXiv

BioHackathon series:
DBCLS BioHackathon 2025
Mie, Japan, 2025
Analytical Workflow Creators

Submitted: 20 Sep 2025

Published by BioHackrXiv.org

DBCLS BioHackathon 2025 report: Creation and Publication Analytical Workflow of Creators' Interests

Ryo Maemda [1], **Hyeokjin Kwon** [2], **Pitiporn Noisagul** [3], **and Sora Yonezawa** [1]

1 Hiroshima University **2** University of Potsdam **3** Center of Multidisciplinary Technology for Advanced Medicine (CMUTEAM), Faculty of Medicine, Chiang Mai University, Chiang Mai, Thailand

Introduction

As part of the DBCLS BioHackathon 2025, we here report about creating and publishing analytical workflow. The analytical workflow is usually based on shell scripts. However, problems of reusability and environmental dependencies are sometimes occuring (Nahan Maligeay, 2024). Here, we aimed to this problems, the workflow based on workflow languages is developed.

Structural variants (SVs) are a major source of genetic variation and can impact disease (including cancer)Beyond 1000 genomes: going deeper and wider. However, traditional analyses use a single linear reference (e.g., GRCh38 or T2T-CHM13) which may miss population-specific sequences and bias read alignment. Recent efforts like the Human Pangenome Reference Consortium (HPRC) and Chinese Pangenome Consortium (CPC) have built pan-genome references that incorporate multiple haplotypes to better represent human diversity (Yang Gao & Xu, 2023). Pangenome graphs include additional structural variants and novel sequences, improving read alignment rates and variant discovery (Maxat Kulmanov & Kawai, 2025). For example, each CPC genome had tens of megabases of sequence not found in GRCh38 or even T2T-CHM13, underscoring how a single reference is incomplete. Using a pan-genome as reference can therefore reduce mapping bias and improve SV detection – studies have shown pangenome-based variant calling finds more variants and higher accuracy than linear references.

Results

Metatranscriptomic analysis

We already published shell scripts on GitHub for metatranscriptomic analysis. Although the software versions used in these shell scripts were listed in thearticle, managing their versions individually can be difficult for users. During DBCLS BioHackathon 2025, the published shell scripts were converted into CWL scripts, and 13 steps are now available on github.com/RyoMameda/workflow_cwl/tree/main/Tools. All scripts work with Docker images.

In addition, we combined the scripts into sub-workflows, each corresponding to different parts of the analysis pipeline: (i) construction of metagenomic contigs and protein prediction, (ii) mapping of metagenomic or metatranscriptomic reads to predicted protein-coding sequences (CDSs), and (iii) gene annotation of predicted CDSs. The workflows are also available on github.com/RyoMameda/workflow_cwl/tree/main/Workflow, and publication on WorkflowHub is in progress (DOI pending approval by the gatekeeper).

We confirmed that the published CWL files work correctly with test datasets (metagenomic reads: SRR27548858; metatranscriptomic reads: SRR27548863, SRR27548864, SRR27548865). The workflow image showing the constructed parts is provided below.

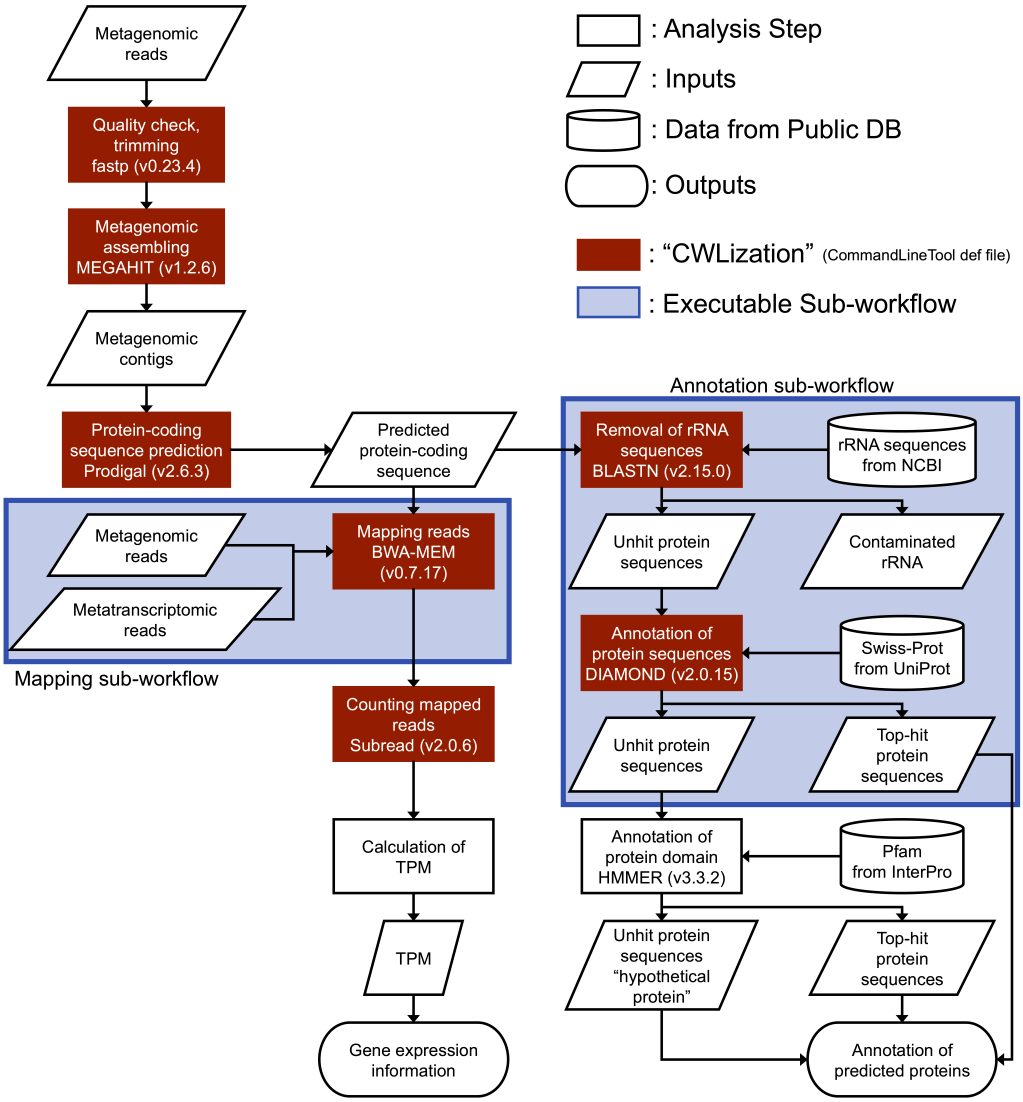

Figure 1: Metatranscriptomic Analysis Workflow

Pangenome-Based SV Calling Benchmark

We selected data from the referenced study and restricted the analysis to the Dai population to serve as the truth dataset. Because access to the original raw reads is delayed, we used the provided CRAM files aligned to both T2T and GRCh38, converting and merging per-sample reads into a single FASTQ for each individual. From each merged FASTQ, we performed random downsampling to approximately 17×, 10×, and 5× relative to the estimated T2T genome size. For mapping, we used the CPC+HPRC+CHM13v2 pangenome graph (attributed to Prof. Shuhua Xu's group) and adopted the Clara Parabricks toolchain: minimap2 for linear-reference alignment and vg giraffe for graph mapping. For SV benchmarking, we used Truvari, deriving a truth VCF by converting the GAF-based SV set provided by the 1KG_ONT_Vienna resource into VCF format (Siegfried Schloissnig & Korbel, 2025) github .

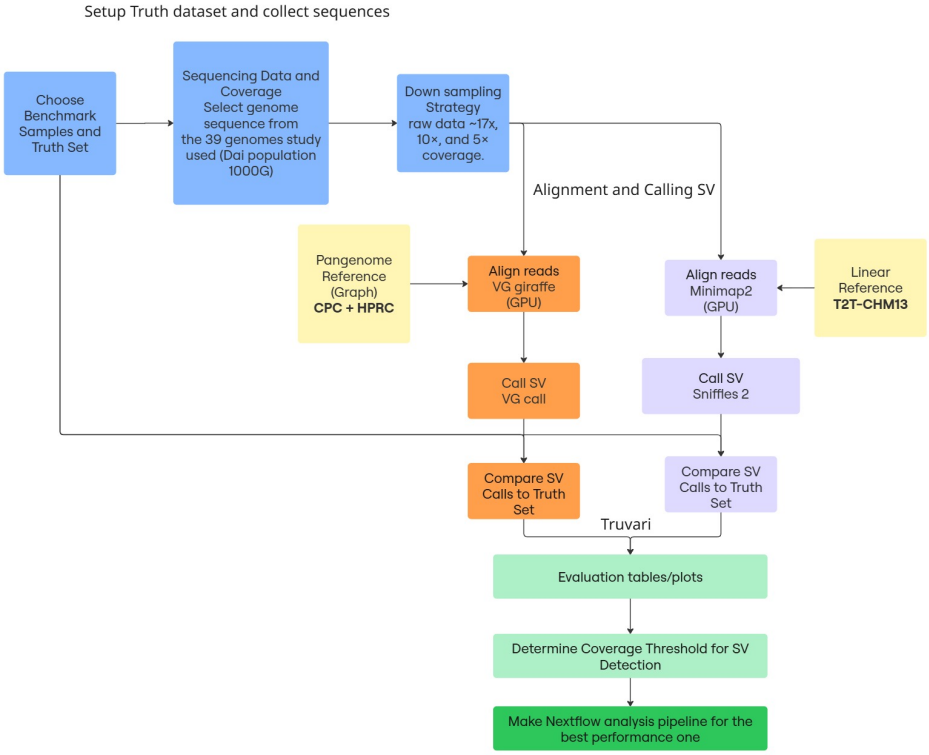

Figure 2: Pangenome-Based SV-Calling Benchmark: Design and Evaluation Flow

Discussion

Consideration to Software Quality

The official website of CWL provides a set of recommended best practices to keep in mind when writing a Common Workflow Language description. Appliying these practices to tools and workflows can improve their software quality. Also, FAIR principles are naturally satisfied by following these practices. Even though more application of these practices is generally better, not all are required.

We evaluated these practices in the perspective of life scientists, who are not necessarily skillful software developers. We classified them into difficulty, importance, and applicability categories. The evaluation is only for this hackathon project, which time and resources are limited. Therefore, this may not be generalized to other cases.

- D - Difficulty (E:Easy, M:Medium, H:Hard)
- I - Importance (L:Low, M:Medium, H:High)
- A - Applicability (Y:Yes, M:Maybe, N:No)

Practice Name	D	I	A	Description
Use class type for files	E	M	Y	Avoid using `type: string` for input/output files. Use `type: File` or `type: Directory` appropriately.
License Declaration	M	H	Y	Include a license field in all tools/workflows. Prefer licenses corresponding to SPDX identifier like Apache 2.0.

Practice Name	D	I	A	Description
Author Attribution	E	M	Y	Include author and contributor information. Use unambiguous identifiers like ORCID.
Software Requirement (dep)	H	H	M	List dependencies using short names under `SoftwareRequirement`.
Software Requirement (ver)	H	H	M	Specify known working tool versions under `SoftwareRequirement`.
SciCrunch Identifiers (RRID)	H	M	M	Include SciCrunch identifiers for dependencies in `https://identifiers.org/rrid/RRID:SCR_NNNNNN` format.
Informative Identifiers	E	H	Y	Use descriptive names for inputs/outputs (e.g., `unaligned_sequences`) instead of generic ones (`fasta1`).
File Format Specification (EDAM)	H	H	N	Specify file formats using identifiers from EDAM (e.g., `format: edam:format_3489`).
Streaming Compatibility	E	L	M	Mark input/output files that are read/written in a streaming compatible way as `streamable: true`.
Single Operation Focus	E	H	Y	Each `CommandLineTool` should focus on a single operation. Avoid overcomplicating with unnecessary options.
Custom Type Definitions	E	H	Y	Define custom types in separate YAML files for reusability.
Top-Level Label & Doc	E	H	Y	Include a short label and, if useful, a longer doc for summarizing the tool/workflow.
Enum Types	E	L	M	Use `type: enum` for elements with a fixed list of valid values.
JavaScript Evaluation	M	M	M	Evaluate the use of JavaScript and consider first use of built-in File properties instead.
Peer Review	H	H	N	Have a colleague test and provide feedback on the tool description.
Subworkflow Feature Requirement	M	H	M	Utilize `SubworkflowFeatureRequirement` for modular workflows with abstractable components.
Container Conformity	M	M	M	Ensure software containers conform to the "Recommendations for the packaging and containerizing of bioinformatics software".

Scalability Considerations

Building the required pangenome graph indices for GPU-accelerated mapping proved time-consuming and storage-intensive. Given the end-to-end data footprint—from FASTQ through

graph indices—we limited the current benchmarking run to a subset of Dai samples. Moreover, there are comparatively few mature tools for calling SVs directly from graph-aligned reads, which constrained our choice of methods. Despite these practical limits, the workflow enables systematic evaluation across decreasing coverages and provides a clear path to expand benchmarking as resources allow.

Next Step

The main workflow of metatranscriptomic analysis could not be fully constructed during this BioHackathon. Further work is needed to complete its publication.

Author Contribution

workflow creation, S.Y., P.N. and R.M.; validation, S.Y., P.N. and R.M.; critical commets, H.K. and S.Y.; writing, R.M., P.N. and H.K..

Acknowledgments

The authors gratefully acknowledge the Bioinformatics Academic Association in Thailand (BAT) for fostering collaboration.

References

Maxat Kulmanov, Y. L., Saeideh Ashouri, & Kawai, Y. (2025). Phased genome assemblies and pangenome graphs of human populations of japan and saudi arabia. *Scientific Data*, *12*(1), Article number: 1316. https://doi.org/10.1038/s41597-025-05652-y **[cito:citation]**

Nahan Maligeay, K. B., Noémie Bossut. (2024). *Why do scientific workflows still break?* https://doi.org/10.1145/3676288.3676300 **[cito:citation]**

Siegfried Schloissnig, J. E., Samarendra Pani, & Korbel, J. O. (2025). Structural variation in 1,019 diverse humans based on long-read sequencing. *Nature*, *644*(7950), 442–452. https://doi.org/10.1038/s41586-025-09290-7 **[cito:citation]**

Yang Gao, H. C., Xiaofei Yang, & Xu, S. (2023). A pangenome reference of 36 chinese populations. *Nature*, *619*(7967), 112–121. https://doi.org/10.1038/s41586-023-06173-7 **[cito:citation]**