An Agent-Based Viral Venture: Adaptive Tool Selection for Scalable Genomics

Naomi Kolodisner
kolodisner@arizona.edu
University of Arizona
Tucson, Arizona, USA

Alok Kamatar, J. Greg Pauloski, Kyle Chard
(advisors)
{alokvk2,jgpauloski,chard}@uchicago.edu
University of Chicago
Chicago, Illinois, USA

Abstract

Fecal Microbial Transplant (FMT) is an effective procedure for restoring gut microbiome balance in patients with *Clostridioides difficile* infection by introducing healthy donor microbes. Tracking viral genomes during FMT provides insight into microbial community transfer and recovery. We developed a viral detection workflow that processes metagenomic samples to identify, dereplicate, cluster, and annotate viral sequences using GeNomad, CheckV, MMseqs2, and BLAST. The workflow links viral sequences to donor and patient samples, enabling longitudinal tracking. Traditionally, such workflows run sequentially with predefined tools and steps. We compare this workflow against an agent-based workflow that selects the viral detection tool dynamically based on the sequence quality and database match-scores of prior samples. Scaling experiments show that parallelizing the workflow using Parsl reduces runtime by over 50%. Tool comparison demonstrates trade-offs in speed, quality, and match ratio, demonstrating the benefits of adaptive, agent-driven workflows for scalable viral detection in microbiome studies.

ACM Reference Format:
Naomi Kolodisner and Alok Kamatar, J. Greg Pauloski, Kyle Chard (advisors). 2025. An Agent-Based Viral Venture: Adaptive Tool Selection for Scalable Genomics. In *Proceedings of The International Conference for High Performance Computing, Networking, Storage, and Analysis (SC '25).* ACM, New York, NY, USA, 2 pages.

1 Introduction

Fecal microbial transplantation (FMT) is a procedure in which stool from a healthy donor is introduced into the intestinal tract of a recipient. FMT is used to treat recurrent infections caused by *Clostridioides difficile* (*C. diff*), a bacterium that disrupts the gut microbiome, enabling overgrowth and disease. By introducing microbes from a healthy donor, FMT can help restore microbiome balance.

Viruses are an understudied but important component of the gut microbiome, particularly in the context of FMT. To investigate their role in FMT-mediated recovery, we developed a viral detection workflow to track viral populations as they move from donor to recipient and change over the course of treatment. We applied this approach to metagenomic datasets from Smillie et al. [Smillie et al. 2018], which include longitudinal stool samples from FMT donors and recipients.

SC '25, St. Louis, MO

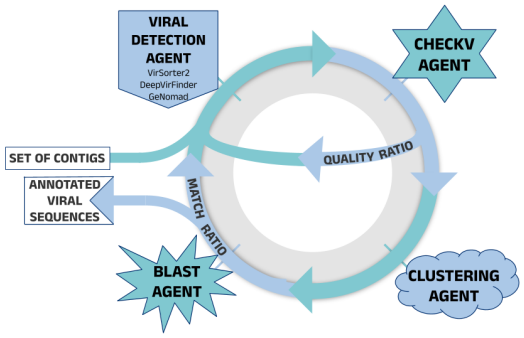

Figure 1: Agentic Workflow with Tool Selection

2 Viral Detection Steps

To detect and annotate viral sequences in a FMT sample, our workflow uses GeNomad, CheckV, MMseqs2, and BLAST. The workflow takes assembled contigs from each sample produces a list of detected viruses, including identifiers that allow tracing to the original patient or donor. The workflow processes as follows:

- **Step 1: Viral Detection (GeNomad)** – Run GeNomad on all metagenomic samples using both machine learning and homology-based methods.
- **Step 2: Quality Assessment (CheckV)** – Evaluate completeness and quality of predicted viral genomes.
- **Step 3: Dereplication (MMseqs2)** – Remove duplicate or highly similar viral sequences across samples.
- **Step 4: Clustering (MMseqs2)** – Combine dereplicated sequences and cluster to select representative sequences.
- **Step 5: Identification (BLAST) 5a:** Split representative sequences into smaller files for efficient BLAST processing; **5b*:** Create a BLAST database using the Aggregated Gut Viral Catalogue (AVrC) [Galperina et al. 2025]. (*one-time setup); **5c:** Compare representative sequences against the AVrC database.
- **Step 6: Consolidation and Annotation – 6a:** Merge BLAST output files into one file; **6b:** Collect annotation info from AVrC and link sequences to original sample IDs.

3 Computational Framework

We use Parsl, a Python library for developing parallel and distributed workflows, to execute viral detection tasks through traditional DAG-based execution, where task order is predefined and

Table 1: Tool Comparison of Match Ratio, Quality and Time

	Match Ratio	Quality Ratio	Time (Hours)
VirSorter2	0.276	0.0034	7.80
DeepVirFinder	0.065	0.0017	1.91
GeNomad	0.434	0.0026	2.18
Epsilon Greedy	0.388	0.0027	2.43

dependencies are resolved statically. While this approach is effective for fixed workflows, it lacks adaptability and flexibility. To address this, we use Academy, a framework for building modular, stateful agents that operate asynchronously and communicate through message passing. Unlike static DAGs, Academy enables persistent state and flexible control logic, allowing the workflow to dynamically adjust tool selection and execution strategies at runtime.

3.1 Adaptive Workflow with Tool Selection

To introduce adaptability into viral detection, we implemented an agent-based workflow where tool selection is guided by an epsilon-greedy algorithm ($\epsilon = 0.6$). At each iteration, the system selects one of three viral detection tools (VirSorter2, DeepVirFinder, GeNomad) based on performance metrics while balancing exploration and optimization. These tools excel in different circumstances because they employ fundamentally different strategies: VirSorter2 relies on homology-based detection, DeepVirFinder uses a machine learning–based approach, and GeNomad combines ML and homology.

Two agents provide key feedback for performance evaluation:

- **CheckV Agent:** Assigns a quality score based on the proportion of detected viral sequences classified as high quality.
- **BLAST Agent:** Assigns a match score based on the proportion of viral sequences matching the reference database.

This feedback loop enables dynamic tool selection that improves quality without compromising efficiency.

4 Evaluation

We evaluate workflow scalability and tool selection efficacy.

4.1 Scalability of the Linear Workflow

We evaluated scaling performance of our workflows by running them with configurations ranging from 16 cores (out of 94 cores) to 8 compute nodes. We compare against a *sequential* baseline which runs each step sequentially. As shown in Figure 2, execution time decreased substantially when scaling to 4 nodes for both the linear and adaptive workflows, dropping from approximately 4 hours to around 2 hours. Beyond four nodes, additional scaling provided minimal benefit, with execution times plateauing near 2 hours and exhibiting low variability across higher node counts. This plateau occurs because the workload is insufficiently large to fully utilize more than four nodes. With 12 samples, the workflow continued to improve up to six nodes before reaching a similar plateau, indicating that a larger workload enables better utilization of additional nodes.

4.2 Adaptive Tool Selection Using Agents

Table 1 compares the quality and match ratios achieved by each tool and the epsilon-greedy strategy which selects tools adaptively.

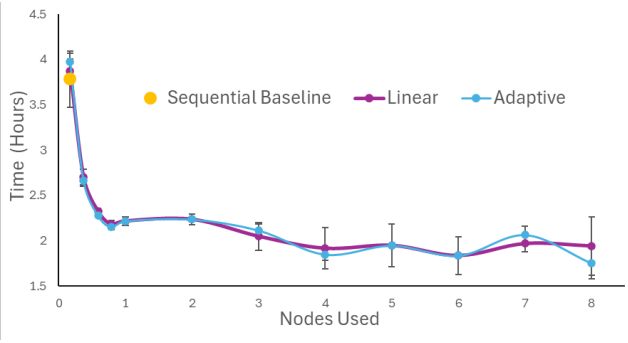

Figure 2: Workflow Scaling with three Samples.

VirSorter2 achieved the highest quality scores, whereas GeNomad provided the best matches to the AVrC database, resulting in the most accurate annotations for our current objectives. Execution time varied between tools. DeepVirFinder was the fastest, followed by GeNomad, while VirSorter2 required the longest runtime.

5 Summary

We developed a scalable viral detection workflow that combines DAG-based parallel execution with agent-driven adaptive tool selection. Using Parsl, the workflow efficiently executes tasks across multiple compute nodes, while the epsilon-greedy agent dynamically selects between VirSorter2, DeepVirFinder, and GeNomad based on quality and match metrics, optimizing the trade-off between quality and annotative abilities.

We will analyze GeNomad-identified viruses, leveraging its high match ratio, to track their transfer and persistence in FMT recovery, also integrating VirSorter2 outputs for higher-quality sequences and exploring additional annotation methods for deeper insights.

References

Yadu Babuji, Anna Woodard, Zhuozhao Li, Daniel S. Katz, Ben Clifford, Rohan Kumar, Lukasz Lacinski, Ryan Chard, Justin M. Wozniak, Ian Foster, Michael Wilde, and Kyle Chard. 2019. Parsl: Pervasive Parallel Programming in Python. *HPDC '19: Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing* (2019), 25–36.

Antonio Pedro Camargo, Simon Roux, Frederik Schulz, Michal Babinski, Yan Xu, Bin Hu, Patrick S. G. Chain, Stephen Nayfach, and Nikos C. Kyrpides. 2024. Identification of mobile genetic elements with geNomad. *Nature Biotechnology* 42 (2024), 1303–1312.

Anastasia Galperina, Gabriele Andrea Lugli, Christian Milani, Willem M. De Vos, Marco Ventura, Anne Salonen, Bonnie Hurwitz, and Alise Jany Ponsero. 2025. The Aggregated Gut Viral Catalogue (AVrC): A unified resource for exploring the viral diversity of the human gut. *PLOS Computational Biology* 21, 5 (2025).

J. Gregory Pauloski, Yadu Babuji, Ryan Chard, Mansi Sakarvadia, Kyle Chard, and Ian Foster. 2025. Empowering Scientific Workflows with Federated Agents. *arXiv:2505.05428v2* (2025).

Kenneth E. Schackart, Jessica B. Graham, Alise J. Ponsero, and Bonnie L. Hurwitz. 2023. Evaluation of computational phage detection tools for metagenomic datasets. *Frontiers in Microbiology* 14 (2023).

Christopher S. Smillie, Jenny Sauk, Dirk Gevers, Jonathan Friedman, Jaeyun Sung, Ilan Youngster, Elizabeth L. Hohmann, Christopher Staley, Alexander Khoruts, Michael J. Sadowsky, Jessica R. Allegretti, Mark B. Smith, Ramnik J. Xavier, and Eric J. Alm. 2018. Strain Tracking Reveals the Determinants of Bacterial Engraftment in the Human Gut Following Fecal Microbiota Transplantation. *Cell Host 'I&' Microbe* 23, 2 (2018), 229–240.