Nov 14, 2024

Benchmarking PathoGFAIR

DOI
https://dx.doi.org/10.17504/protocols.io.e6nvwbp4zvmk/v1

Engy Nasr[1], anna.henger[2], Björn rüning[1], Paul Zierep[1], Bérénice Batut[3]

[1]Albert-Ludwigs-Universität Freiburg; [2]Biolytix AG, Switzerland; [3]Institut Francais de Bioinformatique

Bérénice Batut: Plateforme AuBi, Mésocentre Clermont-Auvergne, Université Clermont Auvergne, 63170 Aubière, France;

GigaScience Press

Engy Nasr
Albert-Ludwigs-Universität Freiburg

DOI: https://dx.doi.org/10.17504/protocols.io.e6nvwbp4zvmk/v1

External link: https://github.com/usegalaxy-eu/PathoGFAIR/tree/main

Protocol Citation: Engy Nasr, anna.henger , Björn rüning, Paul Zierep, Bérénice Batut 2024. Benchmarking PathoGFAIR. **protocols.io** https://dx.doi.org/10.17504/protocols.io.e6nvwbp4zvmk/v1

Manuscript citation:
https://www.biorxiv.org/content/10.1101/2024.06.26.600753v1

Protocol status: Working
We use this protocol and it's working

Created: October 27, 2024

Last Modified: November 14, 2024

Protocol Integer ID: 110986

Keywords: PathoGFAIR, Pathogen Detection, Benchmark, Salmonella, Galaxy, Workflows, pathogen identification in metagenomic dataset, tracking pathogen, similar pathogen identification system, metagenomic dataset, based pathogen identification, benchmarking pathogfair, benchmark pathogfair workflow, pathogen identification, pathogen, terms of the spiked pathogenic strain, spiked pathogenic strain, relevant virulence factor, taxonomy profiling, steps to benchmark, complete analysis pipeline, benchmark, pathoggfair manuscript, oxford nanopore, identifying gene, virulence, pathogfair github repository, pipeline, diagnostic, nanopore

Abstract

PathoGFAIR is a set of easy-to-use workflows for detecting and tracking pathogens in (meta)genomic samples, especially from Oxford Nanopore sequencing. Designed within the Galaxy platform, it offers a complete analysis pipeline; from quality control and taxonomy profiling to identifying genes linked to virulence and resistance. Researchers can use PathoGFAIR for various applications, like food safety, diagnostics, and outbreak tracking, with flexible options to customize the workflows as needed. This protocol presents steps to benchmark PathoGFAIR against similar systems and pipelines.

Objective

To benchmark PathoGFAIR workflows against similar pathogen identification systems/pipelines, ensuring a robust comparison based on their ability to detect pathogens and perform gene-based pathogen identification in metagenomic datasets.

Systems and Pipelines

All systems/pipelines mentioned in **Table 1** of the manuscript are selected for benchmarking.

These pipelines, like PathoGFAIR, perform Gene-based pathogen identification or similar analyses, making them suitable for comparative analysis. This ensures that all systems are tested for their ability to identify pathogens and relevant virulence factors.

Testing Datasets

- **Dataset Source**: The **46 samples without prior isolation** from the PathogGFAIR manuscript (detailed in **Supplementary Table T1** & **PathoGFAIR GitHub repository**). These spiked chicken samples were prepared and sequenced under specific controlled conditions explained in Protocols.io.
- **Why These Datasets**: the metadata of these datasets are well known, in terms of the spiked pathogenic strain, Ct, CFU/ml value, etc.

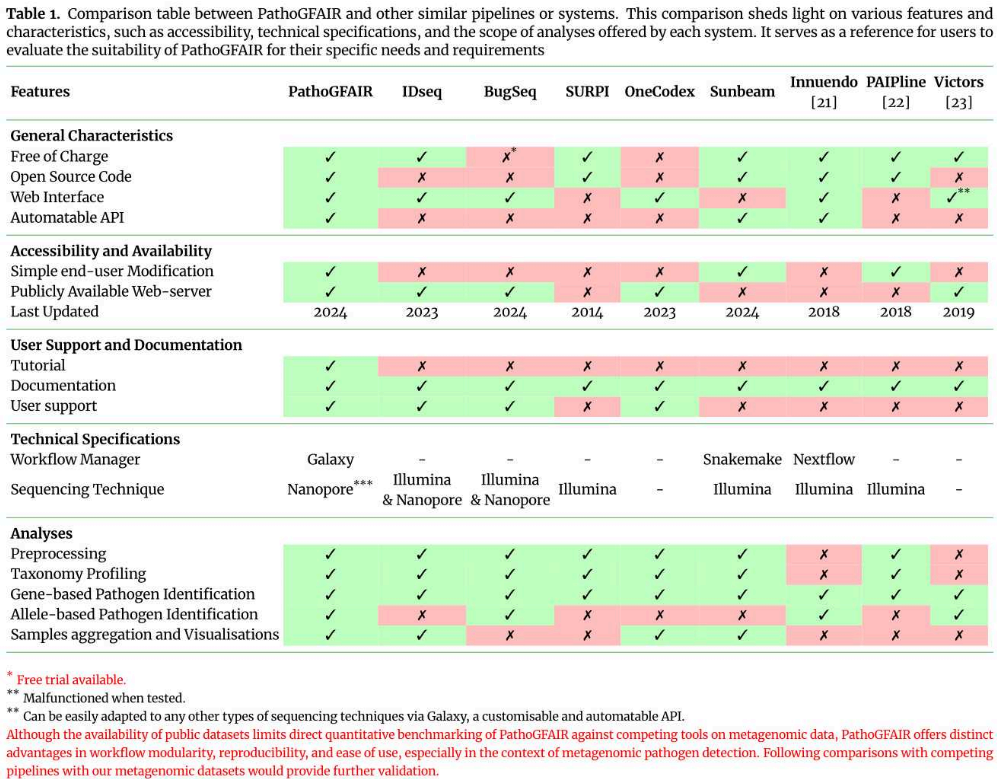

Table 1. Comparison table between PathoGFAIR and other similar pipelines or systems. This comparison sheds light on various features and characteristics, such as accessibility, technical specifications, and the scope of analyses offered by each system. It serves as a reference for users to evaluate the suitability of PathoGFAIR for their specific needs and requirements

Features	PathoGFAIR	IDseq	BugSeq	SURPI	OneCodex	Sunbeam	Innuendo [21]	PAIPline [22]	Victors [23]
General Characteristics									
Free of Charge	✓	✓	✗*	✓	✗	✓	✓	✓	✓
Open Source Code	✓	✗	✗	✓	✗	✓	✓	✓	✗
Web Interface	✓	✓	✓	✗	✓	✗	✓	✗	✓**
Automatable API	✓	✗	✗	✗	✗	✓	✓	✗	✗
Accessibility and Availability									
Simple end-user Modification	✓	✗	✗	✗	✗	✓	✗	✓	✗
Publicly Available Web-server	✓	✓	✓	✗	✓	✗	✗	✗	✓
Last Updated	2024	2023	2024	2014	2023	2024	2018	2018	2019
User Support and Documentation									
Tutorial	✓	✗	✗	✗	✗	✗	✗	✗	✗
Documentation	✓	✓	✓	✓	✓	✓	✓	✓	✓
User support	✓	✓	✓	✗	✓	✗	✗	✗	✗
Technical Specifications									
Workflow Manager	Galaxy	-	-	-	-	Snakemake	Nextflow	-	-
Sequencing Technique	Nanopore***	Illumina & Nanopore	Illumina & Nanopore	Illumina	-	Illumina	Illumina	Illumina	-
Analyses									
Preprocessing	✓	✓	✓	✓	✓	✓	✗	✓	✗
Taxonomy Profiling	✓	✓	✓	✓	✓	✓	✗	✓	✗
Gene-based Pathogen Identification	✓	✓	✓	✓	✓	✓	✓	✓	✓
Allele-based Pathogen Identification	✓	✗	✓	✗	✗	✗	✓	✗	✓
Samples aggregation and Visualisations	✓	✓	✗	✗	✓	✓	✗	✗	✗

* Free trial available.

** Malfunctioned when tested.

** Can be easily adapted to any other types of sequencing techniques via Galaxy, a customisable and automatable API.

Although the availability of public datasets limits direct quantitative benchmarking of PathoGFAIR against competing tools on metagenomic data, PathoGFAIR offers distinct advantages in workflow modularity, reproducibility, and ease of use, especially in the context of metagenomic pathogen detection. Following comparisons with competing pipelines with our metagenomic datasets would provide further validation.

Table 1 - Comparison table between PathoGFAIR and other similar pipelines or systems. This comparison sheds light on various features and characteristics, such as accessibility, technical specifications, and the scope of analyses offered by each system. It serves as a reference for users to evaluate the suitability of PathoGFAIR for their specific needs and requirements, from PathoGFAIR Manuscript.

Troubleshooting

Benchmarking Steps

1 **Selection of Benchmarking Pipelines:**

Goal: Identify systems/pipelines for benchmarking based on their availability, compatibility with metagenomic sequencing, and relevance for pathogen identification.

Actions:

- **Availability Check**: Review each system/pipeline for accessibility and current functionality:
- **Free Access**: Available at no cost.
- **Free Trial**: Usable with a trial period or partial free access.
- **Paid Access Only**: Available solely through subscription or payment.
- **Non-functional**: Outdated, discontinued, or malfunctioning.
- **Sequencing Compatibility Check**: Confirm each system/pipeline's compatibility with single-end Nanopore metagenomic data.
- **Pathogen Identification Capability**: Assess each pipeline's features and compatibility for pathogen identification, focusing on their utility for benchmarking PathoGFAIR.

Exclusion Criteria: Exclude pipelines that are inaccessible, paid-only, non-functional, or incompatible with Nanopore metagenomic data.

Outcome: Generate a list of selected pipelines, each with a clear rationale for inclusion based on accessibility, compatibility, and relevance to pathogen identification. This will form the basis for an objective benchmark comparison.

2 **Pipeline Usage and Setup:**

Goal: Set up and run the chosen systems/pipelines to ensure they can be easily executed and reproduced.

Action:
Check how each system operates:

- Is it command-line-based or platform-based (web interface)?
- Can it be automated via a script or containerised environment?

Outcome: A guide in the PathoGFAIR Github repository for running the benchmark.

3 **Running the Benchmark:**

Goal: Use **46 Samples Without Prior Pathogen Isolation** described in PathoGFAIR manuscript to run the benchmark and compare results across systems.

Actions:

- Load the 46 samples into each selected system.
- Record the results of pathogen identification for each sample at different taxonomic levels (genus, species, subspecies, strain).
- Identify how each system performs relative to the expected pathogen as stated in the sample metadata.

Outcome: Collection of all results for each system across all 46 samples.

4 **Collecting Results and Handling Taxonomic Resolution Discrepancies:**

Goal: Standardise results collection and comparison, especially considering differences in taxonomic resolution between pipelines.

Actions:

- Record the pathogens detected by each system per sample.
- Create a table documenting the taxonomic rank at which each system was able to detect the expected pathogen:
  - **0** = Expected pathogen was not detected
  - **1** = Detected at genus level
  - **2** = Detected at species level
  - **3** = Detected at subspecies level
  - **4** = Detected at strain level
- If a system cannot detect to a specific rank (e.g., only detects to the species level), this will be factored into the final scoring.

Outcome: A comprehensive results table showing how each pipeline performs across taxonomic ranks.

5 **Performance Evaluation:**

Goal: Evaluate how each system performs in terms of available known specs e.g. runtime, memory usage, and scalability.

Action:

For each system, check for performance if existed:

- **Runtime** (how long it takes to complete pathogen identification)
- **Memory usage** (how much computational memory is required)
- **Scalability** (can it handle large datasets effectively?)

Outcome: A summary of the performance of each system, offering insight into its efficiency and scalability, this summary is explained now in the manuscript.

6 **Data Visualization and Summary:**

Goal: Summarise and visualise the results from all systems in a comprehensive format for easy interpretation.

Actions:

- Create a **heatmap** using the results table from Step 4, illustrating at which taxonomic level each system detected the expected pathogen.
- Provide a **summary table** of sensitivity/specificity evaluations.
- Highlight the strengths and weaknesses of each system.

Outcome: A clear, visual representation of how each system performed in pathogen identification.

7 **Publishing and Documentation:**

Goal: Ensure the benchmark process is reproducible and accessible for others.

Actions:

- Publish the **benchmark protocol** on **Protocol.io** for public access.
- Upload the **benchmarking guide** to PathoGFAIR's **GitHub repository**, allowing others to replicate the benchmarking process.

Outcome: Complete transparency and reproducibility of the benchmark, ensuring others can validate or extend the findings.

Sensitivity and Specificity of PathoGFAIR

8 **Goal**: Assess how sensitive and specific PathoGFAIR is when detecting the expected pathogen at subspecies rank and avoiding false positives.

Actions: Evaluate sensitivity (true positives) and specificity (false positives) for PathoGFAIR by:

- Comparing results with known spiked pathogens in the dataset.
- Assessing false positives, if any.

Outcome: A sensitivity and specificity evaluation that indicates how accurate PathoGFAIR is in pathogen identification.

Expected Deliverables

9

1. **Results Table**: Detailing pathogen identification levels (genus, species, subspecies, strain) across systems for each sample. Table is available inside the **data/benchmark** directory of the PathoGFAIR GitHub repository.
2. **Heatmap**: Visual representation of pathogen detection success across all systems. Notebook to regenerate the heatmap is available inside the **bin directory** of the PathoGFAIR GitHub repository. The **heatmap** is presented and explained in the manuscript.
3. **Performance Summary**: Including runtime, memory usage, and scalability. Performance summary of all tools is explained in the manuscript.
4. **Sensitivity/Specificity Evaluation**: Comparative accuracy assessment for PathoGFAIR available as a table in the **data/benchmark** directory of the PathoGFAIR GitHub repository and explained in the manuscript.
5. **Published Protocol**: This protocol on **Protocol.io**.
6. **Automated Benchmark Guide**: available on the PathoGFAIR GitHub repository.