Bilkent University
Department of Computer Engineering

Senior Design Project
T2504
Pathogenius

Project Specification Document

Nazlı Apaydın 22202104
Ege Ateş 22201914
Yiğit Ali Doğan 22202329
Yunus Günay 22203758
Ata Uzay Kuzey 22203050

Supervisor: Can Alkan
Innovation Expert: Can Alkan

28.11.2025

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfilment of the requirements of the Senior Design Project course CS491/2.

Contents

1 Introduction

Advances in sequencing technologies have made it possible to analyze genetic material directly from clinical or environmental samples. However, the software tools required to interpret such data are often complex, computationally demanding, and designed for laboratory environments equipped with high-performance servers or cloud-based resources. As a result, many field settings lack practical access to metagenomic analysis, even when rapid identification of potential pathogens could support timely decision-making.

Pathogenius aims to address this limitation by providing a portable metagenomic analysis platform that can operate on modest local hardware without relying on an internet connection. The system processes raw long-read sequencing data, applies preprocessing and species-level classification, and presents the results through an accessible user interface suitable for non-expert users. By combining some of the existing bioinformatics tools within a structured workflow, Pathogenius seeks to make genetic analysis more accessible in settings where traditional infrastructure may not be available.

This document presents the project description, high-level system architecture, development constraints, relevant engineering standards, professional and ethical considerations, design requirements, and feasibility assessments.

1.1 Description

Pathogenius is a portable metagenomic analysis platform designed to identify pathogenic organisms from raw sequencing data in resource-limited clinical or emergency settings. Instead of depending on high-performance compute clusters or cloud-based pipelines, Pathogenius operates on modest local hardware such as laptops or embedded devices. The system processes long-read FASTQ data and delivers clear, species-level diagnostic output, rather than full taxonomic hierarchies, through a user-friendly, locally running interface. Input reads will be treated as Nanopare-style [1] sequences; however, since an actual device is not available, realistic test data can be generated using simulators such as the *icarus* simulator [2].

Pathogenius uses Kraken2, a fast and memory-efficient k-mer-based classifier that determines the most probable species origin of each read. This species-level reporting aligns with the practical needs of clinicians and field workers. Meanwhile, raw FASTQ reads are preprocessed.

Kraken2 then compares these cleaned reads against a reference dataset of complete genomes, where each bacterial or viral species is stored in a separate FASTA file, enabling the system to infer the species composition of the sample.

All stages of the analysis pipeline will be operated using Snakemake, a workflow management system that ensures deterministic execution and modular design. This choice also enables maintainability, scalability, and straightforward integration of the external tools mentioned above. Thus, the first phase of the project focuses on workflow construction and user interface development, while CUDA-level optimization will be explored in later stages. Overall, Pathogenius aims to provide an offline-capable, accessible diagnostic tool that can operate on simulated or real sequencing data, ultimately supporting rapid pathogen detection in real-world medical and crisis-response scenarios.

1.2 High Level System Architecture & Components of Proposed Solutions

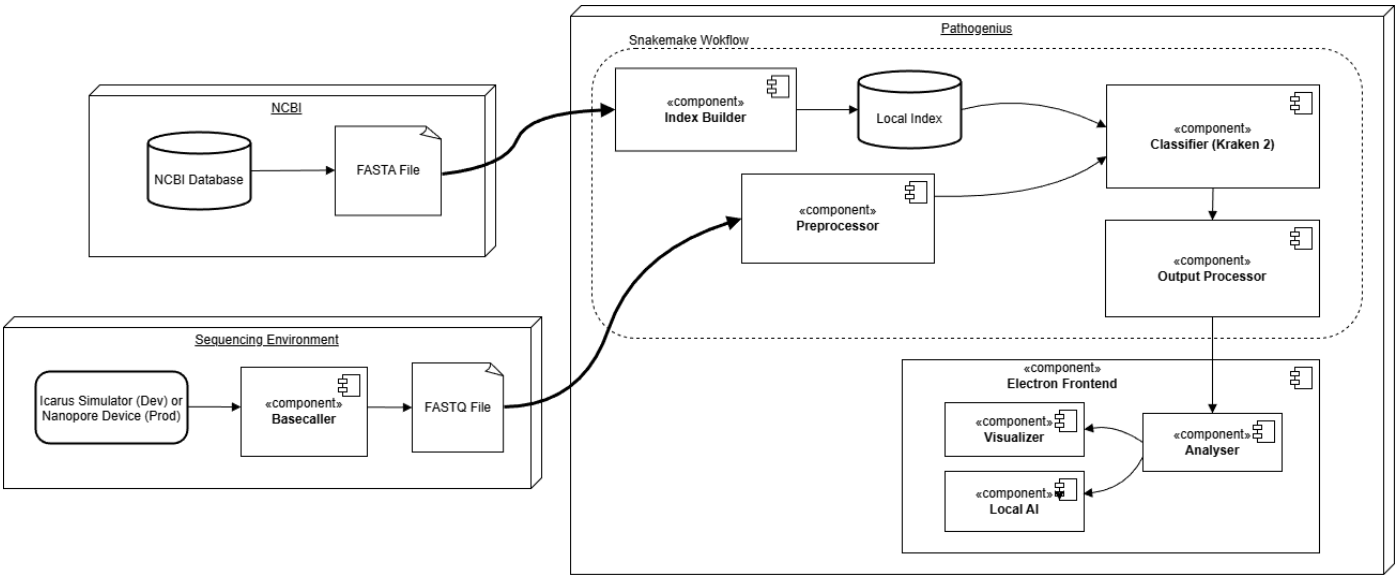

Fig. 1. High Level System Architecture of the Pathogenius and Its Components

The high level architecture of Pathogenius is depicted in Figure 1. The system is composed of discrete modules for the user interface, workflow management and data management which operate fully within the local environment to ensure offline functionality of core systems. The

specific responsibilities of the system components are further detailed in the following subsections.

1.2.1 Sequencing Environment Layer

Generates the raw long-read FASTQ data that serves as input to the analysis workflow.

Components:

- **Icarus Simulator / Nanopare Device:** Produces the electrical signals representing DNA passages through nanopares.
- **Basecaller:** Converts raw electrical signals into nucleotide sequences with associated quality scores, producing a FASTQ file.

Data Flow:

Simulator / Nanopare Device → Basecaller → FASTQ File → Preprocessor (Workflow Layer)

1.2.2 Reference (NCBI) Layer

Provides a genomic reference database and produces the Kraken2-compatible local index required for classification.

Components:

- **NCBI Reference Database:** Publicly available genomes retrieved in FASTA format. These files are stored locally to enable fully offline analysis.

Data Flow:

NCBI FASTA Files → Index Builder (Workflow Layer)

1.2.3 Workflow Layer

Coordinates all analysis steps in a reproducible, modular pipeline, ensuring offline execution and deterministic results.

Components:

- **Preprocessor:** Performs quality control on FASTQ input (filtering, trimming, cleanup) to produce cleaned reads.
- **Index Builder:** A Snakemake-managed workflow step that constructs the Local Index from curated FASTA files.
- **Local Index:** A persistent Kraken2 k-mer database generated from reference genomes.
- **Classifier (Kraken2):** Assigns each cleaned read to most likely species using Local Index.
- **Output Preprocessor:** Aggregates classification output, computes confidence metrics, and prepares structured result files for the Frontend Layer.

Data Flow:

FASTQ File → Preprocessor → Cleaned Reads

FASTA Files → Index Builder → Local Index

Cleaned Reads + Local Index → Classifier → Output Preprocessor → Frontend Layer

1.2.4 Frontend Layer

Provides a locally running, user-friendly interface that allows users to identify analyses, monitor pipeline execution, and interpret species-level results.

Components:

- **Analyser:** Receives processed output from the Workflow Layer and converts it into structured, interpretable result objects.
- **Visualizer:** Displays species-level results, confidence values, and quality metrics generated from the Analyser's output.
- **Local AI (Artificial Intelligence):** An offline model that generates readable explanations and summaries from the Analyser's structured results to support non-expert users.

Data Flow:

Output Preprocessor → Analyser → Visualizer

Output Preprocessor → Analyser → Local AI

1.3 Constraints

1.3.1 Implementation Constraints

- The version controlling and collaboration of the project will be managed through Git and hosted on Github.
- The bioinformatics analysis pipeline will be implemented using Snakemake workflow management to ensure modularity, reproducibility and fault tolerance.
- The graphical user interface (GUI) will be developed with Electron.js to provide a web-like user experience that functions offline, while ensuring cross-platform compatibility [3].
- The system follows a phased development approach. The primary classification engine is currently constrained to the CPU-based Kraken2 algorithm to ensure baseline functionality. However, the underlying hardware is constrained to support NVIDIA CUDA, allowing for the planned migration to GPU-accelerated kernels in later development stages.
- Any implementation of an AI helper assistant is constrained to operate locally. The system must utilize a Small Language Model (SLM) or a quantized LLM that runs directly on the host device to maintain data privacy and offline capability.
- The system is constrained to operate entirely without internet access during the analysis phase. All dependencies, including the frontend assets, reference databases, and analysis tools, must be bundled locally or downloadable prior to field deployment.
- The system must accept raw sequencing data in standard FASTQ format. In the absence of a physical Oxford Nanopore device during development, the system functionality is constrained to using simulated reads generated by the Icarus simulator.

1.3.2 Economic Constraints

- The system is constrained to operate on modest local hardware, such as mid-range commercial laptops, rather than requiring expensive High-Performance Computing (HPC) clusters or cloud-based resources.
- To ensure accessibility in resource-limited settings, the software is constrained to be free of charge. It must not rely on paid cloud APIs (e.g., AWS, OpenAI API) or recurring subscription services that would incur per-use costs.

1.3.3 Ethical Constraints

- Clinical samples may contain human host DNA. The system is ethically constrained to process data transiently on the local device, ensuring that sensitive human genomic information is never exposed.
- The system operates under the constraint of being a decision support system, not a diagnostic device. The user interface must clearly present results as probabilistic evidence.

1.3.4 Environmental Constraints

- Unlike competitor systems that rely on energy-intensive High-Performance Computing (HPC) clusters or continuous cloud connectivity, the system is constrained to minimize its environmental footprint by performing analysis locally on low-power hardware, maximizing performance-per-watt to preserve battery life in off-grid scenarios.

1.3.5 Usability Constraints

- The system is constrained to be usable by field personnel who may lack bioinformatics expertise. The interface must abstract the command-line operations, preventing the user from needing to interact with the underlying terminal or scripts.

1.4 Professional and Ethical Issues

Issue 1: Users without bioinformatics expertise may overinterpret the system's output as fully definitive. Because metagenomic classification is inherently probabilistic and depends on data quality, reference database completeness, and preprocessing accuracy, users may assume that all detected species are equally reliable. Species identified from very few reads, however, may be misinterpreted as clinically significant findings, potentially leading to inappropriate medical decisions.

> **Mitigation:** The interface presents species-level results together with confidence indicators and explanatory notes clarifying that the system provides supportive evidence rather than definitive diagnosis. Results are communicated with appropriate uncertainty markers, and low-confidence findings are visually distinguished or deprioritized.

Issue 2: Inaccurate or low-confidence pathogen identification may mislead users and result in inappropriate treatment decisions.

> **Mitigation:** The system restricts its diagnostic output to species-level identification and employs validated preprocessing tools together with a curated Kraken2 reference database to improve classification accuracy and reduce the likelihood of misleading results.

Issue 3: Clinical metagenomic samples frequently contain substantial amounts of host-derived DNA alongside pathogen DNA. Since sequencing devices read all genetic material in the sample without distinction, raw FASTQ files may include human genomic fragments. This introduces ethical and privacy concerns, as human DNA constitutes personal information that must be handled with strict confidentiality.

> **Mitigation:** All analysis in Pathogenius is conducted completely offline on the user's local machine, ensuring that human genomic fragments never leave the controlled environment. No data is uploaded to external servers, and logs or reports do not store identifiable sequence information.

Issue 4: The system operates exclusively at the DNA sequence level and does not perform protein-level or functional genomic analysis. As a result, clinically relevant features such as antibiotic resistance genes, virulence factors, or specific metabolic pathways cannot be identified.

> **Mitigation:** The documentation and user interface clearly state that the system provides species-level identification only. Users are informed about the tool's diagnostic scope and are encouraged to incorporate supplementary laboratory tests when functional or phenotypic analysis is required. The current prototype explicitly restricts its outputs to avoid misinterpretation.

Issue 5: Kraken2 relies on a locally stored reference database built from publicly available genomes (NCBI). However, reference databases are inherently incomplete, and newly emerging, rare, or poorly characterized organisms may not be represented. If a species present in the FASTQ dataset is missing from the reference database, its reads may be misclassified or remain unclassified, potentially reducing diagnostic accuracy or leading to overlooked pathogens.

> **Mitigation:** Pathogenius uses curated and regularly updated reference datasets to minimize missing taxa, and documentation explicitly notes that unidentified reads may reflect gaps in existing genomic databases rather than their absence from the sample.

1.5 Standards

1.5.1 IEEE 830

The development of Pathogenius follows the principles outlined in the IEEE 830 Software Requirements Specifications (SRS) standard. IEEE 830 provides a systematic framework for defining and documenting both functional and non-functional requirements in a clear, verifiable, and unambiguous manner. The standard's emphasis on clarity and completeness helps prevent misinterpretation among team members, supports accurate communication with supervisors, and provides a solid foundation for validation and testing

during later stages of development. Ultimately, IEEE 830 contributes to producing a well-defined, traceable, and maintainable specification for the Pathogenius platform.

1.5.2 ISO/IEC 25010

The ISO/IEC 25010 [4] standard provides a comprehensive quality model used to evaluate and maintain software product quality throughout the development lifecycle. This model defines a set of high-level quality characteristics, such as performance efficiency, reliability, usability, security, maintainability, and compatibility, which guide the assessment of whether a software system meets user expectations and operational requirements. In the context of Pathogenius, ISO/IEC 25010 serves as a reference framework for ensuring that the platform functions reliably in resource-constrained settings and produces results that users can trust. Therefore, the project aligns its software quality goals with internationally recognized guidelines.

1.5.3 UML 2.5.1 - Unified Modeling Language

Unified Modeling Language (UML) 2.5.1 [5] is employed in the design and documentation of Pathogenius to describe the system's structure, behavior, and component interactions using standardized graphical representations. Within the project, it is used to produce high-level diagrams such as use-case diagrams illustrating user interactions, or activity diagrams describing workflow execution through tools like Kraken2. By adhering to the UML specification, the team ensures that architectural decisions are documented systematically and can be easily understood, reviewed, and maintained. This contributes to clearer design discussions with more accurate implementation.

1.5.4 ISO 9241-210

ISO 9241-210 [6] emphasizes designing software by prioritizing the needs, capabilities, and limitations of its intended users, ensuring that systems are both usable and useful in real-world operational contexts. It also supports iterative evaluation and refinement of the interface. In the context of Pathogenius, this standard is highly relevant because the system is

intended to be used by people who may not possess bioinformatics expertise. Therefore, ISO 9241-210 encourages the development of a clear, intuitive, and error-reducing interface that supports correct interpretation of species-level outputs without overwhelming the user with technical complexity.

2 Design Requirements

2.1 Functional Requirements

2.1.1 Main Features

2.1.1.1 Analysis Management

- **FR-AM-001:** The system shall store all completed analyses in a persistent local database.
- **FR-AM-002:** The system shall display a list of previous analyses showing analysis name, date, input file, and processing status.
- **FR-AM-003:** The system shall allow users to assign custom names and descriptions to analyses.
- **FR-AM-004:** The system shall provide search and filter capabilities for previous analyses.
- **FR-AM-005:** The system shall enable users to reopen completed analyses to view results.
- **FR-AM-006:** The system shall allow users to delete analyses with confirmation prompts.
- **FR-AM-007:** The system shall support exporting and archiving multiple analyses.
- **FR-AM-008:** The system shall use a large language model to convert analysis results into clear, easy-to-read paragraphs.

2.1.1.2 FASTQ Processing

- **FR-FP-001:** The system shall accept FASTQ format files (compressed and uncompressed) as input.

- **FR-FP-002:** The system shall validate input files and display file metadata before processing.
- **FR-FP-003:** The system shall calculate and report confidence scores for each species identification.
- **FR-FP-004:** The system shall execute the analysis pipeline using Snakemake workflow management.
- **FR-FP-005:** The system shall support checkpoint and resume functionality for interrupted analyses.
- **FR-FP-006:** The system shall generate preprocessing and classification quality reports.
- **FR-FP-007:** The system shall support batch processing of multiple FASTQ files.

2.1.1.3 Dataset Management

- **FR-DM-001:** The system shall maintain a local reference database of complete pathogen genomes in FASTA format.
- **FR-DM-002:** The system shall provide functionality to update the reference database when new pathogen genomes become available.
- **FR-DM-003:** The system shall display information about the current database version including the number of species, last update date, and database size.
- **FR-DM-004:** The system shall support manual database curation, allowing users to add, remove, or modify species entries.

2.1.1.4 Notifications

- **FR-NT-001:** The system shall notify users when analysis processing is complete.
- **FR-NT-002:** The system shall alert users when errors occur during processing with descriptive messages.
- **FR-NT-003:** The system shall display notifications for long-running operations with progress updates.

- **FR-NT-004:** The system shall maintain a notification history accessible through the interface.

2.1.1.5 User Registration and Authentication

- **FR-UA-001:** The system shall provide user registration functionality allowing new users to create accounts with username and password.
- **FR-UA-002:** User authentication is not mandatory. The system will support a 'Guest Mode' for immediate offline use. Authenticated users shall gain access to their local history.
- **FR-UA-003:** The system shall authenticate users through a login interface with secure credential validation.
- **FR-UA-004:** The system shall store user credentials securely using encryption.
- **FR-UA-005:** The system shall maintain separate user workspaces ensuring data privacy between users.
- **FR-UA-006:** The system shall accommodate multiple users in a single device with encryption to ensure that the users can keep their analysis secure.

2.1.2 Secondary Features

2.1.2.1 Realtime Data Display

- **FR-RT-001:** The system shall display real-time progress indicators showing current workflow stage and completion percentage.
- **FR-RT-002:** The system shall show live updates of species identifications as reads are classified.
- **FR-RT-003:** The system shall display real-time resource utilization including CPU, memory, and GPU usage.
- **FR-RT-004:** The system shall provide estimated time remaining for ongoing analyses.
- **FR-RT-005:** The system shall update classification statistics dynamically as processing progresses.

- **FR-RT-006:** The system shall display the number of reads processed per minute during analysis.
- **FR-RT-007:** The system shall show preliminary results for completed workflow stages while subsequent stages continue processing.

2.1.2.2 GPU Acceleration

- **FR-GPU-001:** In later stages the system shall detect available NVIDIA CUDA-compatible GPUs on the host system.
- **FR-GPU-002:** The system shall provide an option to enable or disable GPU acceleration for classification tasks.
- **FR-GPU-003:** The system shall automatically optimize workload distribution between CPU and GPU when acceleration is enabled.
- **FR-GPU-004:** The system shall fallback to CPU-only processing if GPU acceleration fails or is unavailable.
- **FR-GPU-005:** The system shall display GPU utilization metrics during accelerated processing.
- **FR-GPU-006:** The system shall support CUDA-accelerated k-mer matching operations for improved performance.

2.2 Non-Functional Requirements

These non-functional requirements define the quality attributes, performance constraints and operational interfaces of the project, Pathogenius. These requirements are categorized according to the ISO/IEC 25010 [4] software quality standard mentioned in section 1.5.2 to ensure the comprehensive coverage and thorough description of the system characteristics.

2.2.1 Usability

- **NFR-USE-01:** The system shall provide a completely graphical user interface (GUI) for all tasks, requiring no interaction with the command-line interface.

- **NFR-USE-02:** The system will display real-time visual feedback, including progress bars and stage completion indicators, during the execution of long-running workflows.
- **NFR-USE-03:** The output dashboard shall visually distinguish between high and low confidence species identifications and other relevant information to support accurate interpretation.
- **NFR-USE-04:** The system shall display descriptive and actionable error messages to the user for efficient error detection and debugging.

2.2.2 Reliability

- **NFR-REL-01:** The system shall perform all core functions, including preprocessing, computation and classification, without an active internet connection.
- **NFR-REL-02:** The installation package shall bundle all necessary dependencies, databases, and analysis tools (e.g. Kraken2) to ensure autonomous operation.
- **NFR-REL-03:** The system shall handle corrupt or truncated FASTQ entries by logging the specific error, skipping the malformed read, and continuing to process the remainder of the file without termination.
- **NFR-REL-04:** The system shall ensure deterministic execution, producing identical output given identical input data and databases.
- **NFR-REL-05:** The system shall strictly treat original input files (raw FASTQ files) as read only and not modify or delete them in any way during any stage of execution.

2.2.3 Performance

- **NFR-PER-01:** The system shall operate on standard consumer-grade hardware.
- **NFR-PER-02:** The system shall provide a mechanism to restrict the memory usage of the Kraken2 or given classifier to prevent exceeding the host machine's available memory.
- **NFR-PER-03:** The system shall complete the preprocessing and classification of a FASTQ file containing simulated reads without introducing significant latency that would hinder the diagnostic interpretation on the minimum specified hardware.

- **NFR-PER-04:** The system shall execute the computationally intensive tasks as background processes to ensure that the user interface remains responsive to the user during the operation.

2.2.4 Supportability

- **NFR-SUP-01:** The analysis workflow shall be implemented with modularized Snakemake rules to pave the way for the independent update and replacement of individual tools without the need for architectural restructuring of the system.
- **NFR-SUP-02:** The system shall generate persistent, human readable logs for analysis sessions, recording analysis parameters, timestamps and other relevant information for troubleshooting purposes.
- **NFR-SUP-03:** The system shall be deployable as a containerized application or managed environment to ensure consistent behavior across different platforms like Windows or Linux.

2.2.5 Scalability

- **NFR-SCA-01:** The system shall support the processing of FASTQ files ranging from 1 MB to 10 GB in size, limited by the host machine's available storage, available memory and when applicable the GPU video memory.
- **NFR-SCA-02:** The system shall automatically identify available resources upon initialization including multiple core CPUs and CUDA enabled GPUs.
- **NFR-SCA-03:** The classification module shall be designed to utilize the most efficient available hardware, multiple threaded CPU processing or GPU accelerated processing when applicable.
- **NFR-SCA-04:** The system shall provide an interface for the user to import valid FASTA files into the reference database, triggering the required indexing process to detect new pathogens without requiring software code updates.

3 Feasibility Discussions

This section evaluates the viability of Pathogenius by examining both the commercial landscape and academic research foundations. The market analysis identifies existing pathogen detection solutions and establishes how Pathogenius differentiates itself through its combination of unbiased metagenomic analysis, offline operation, and modest hardware requirements. The academic analysis reviews the scientific literature supporting the technical approach, validates the chosen computational methods, and identifies potential challenges that must be addressed during development. Together, these analyses demonstrate that Pathogenius addresses a genuine market need with technically sound methods while operating within realistic constraints for a portable diagnostic platform.

3.1 Market & Competitive Analysis

The market for portable pathogen detection has seen significant growth in recent years, driven by the need for rapid diagnostics in resource-limited settings, emergency response scenarios, and point-of-care applications. An examination of existing commercial solutions and competing technologies reveals both the opportunities and differentiation points for Pathogenius.

3.1.1 Commercial Detection Kits

Bio-Rad's IQ-Check kits represent a widely adopted approach to pathogen detection, particularly for water sample analysis [7]. These kits employ straightforward chemical assays to identify specific pathogens such as Salmonella II and Listeria species. While the simplicity of these kits makes them accessible to users without specialized training, their diagnostic scope is inherently limited: each kit targets a single pathogen species, requiring practitioners to anticipate which organisms they are testing for before sample collection. In contrast, Pathogenius employs an unbiased metagenomic approach that simultaneously screens for multiple pathogenic organisms without requiring prior knowledge of the sample composition. This fundamental difference positions Pathogenius as a discovery tool rather than a confirmatory test, making it more suitable for scenarios where the pathogen identity is unknown.

Similarly, Norgen Biotek offers a range of waterborne pathogen detection kits validated for use with commercial PCR instruments including the Qiagen Rotor-Gene Q, BioRad CFX96 Touch, and QuantStudio 7 Pro systems [8]. These kits are marketed for research applications and, in some cases, carry CE marking for in vitro diagnostic use. However, they share the same operational constraint as the IQ-Check products: they require specific primers designed for predetermined target organisms. Furthermore, these PCR-based approaches necessitate access to thermal cycling equipment and maintain a dependency on reagent supply chains. Pathogenius diverges from this model by processing raw sequencing data through computational methods, thereby reducing ongoing consumable costs once the initial reference database is established. The trade-off is increased computational complexity and longer processing time compared to rapid PCR assays.

3.1.2 Academic and Research Tools

The academic landscape reveals several projects with overlapping goals but distinct implementation strategies and target applications. ETH Zurich developed an on-site airborne pathogen detection system focused on infection risk mitigation in indoor environments, successfully identifying SARS-CoV-2 variants alongside lethal bacterial and fungal species [9]. This project prioritized innovations in aerosol sampling and biosensing strategies rather than the computational analysis pipeline itself. While both systems address real-time pathogen identification, the ETH Zurich work targets environmental monitoring of air quality, whereas Pathogenius focuses on clinical sample analysis for diagnostic support. The sampling methodologies and sample preparation workflows differ substantially between these applications.

Nanometa Live represents perhaps the closest academic analogue to Pathogenius [10]. Developed specifically for Oxford Nanopore sequencing data, Nanometa Live combines a backend analysis pipeline with a frontend graphical user interface to deliver real-time metagenomic insights as sequencing data is generated [11]. The system incorporates BLAST validation as a secondary confirmation step, enhancing classification confidence at the cost of increased computational demand and longer processing times. This validation approach requires access to comprehensive sequence databases and sufficient computational resources to perform alignment-based searches. Pathogenius differentiates itself through its emphasis on

portability and modest hardware requirements: by relying primarily on the k-mer-based Kraken2 classifier without computationally expensive alignment validation, the system can operate on standard laptop hardware. This design decision prioritizes accessibility and deployment flexibility over the highest possible classification accuracy, which is appropriate for the intended use case of field diagnostics where some uncertainty is acceptable as long as it is clearly communicated.

Another relevant academic effort applied Oxford Nanopore MinION sequencing with GraphMap for comprehensive pathogen detection in potato field soil [12]. This agricultural pathogen surveillance project demonstrates the versatility of metagenomic approaches across different sample types and application domains. However, the target users, operational requirements, and performance criteria differ markedly from clinical pathogen detection. Agricultural applications may tolerate longer turnaround times and operate under different regulatory frameworks compared to clinical diagnostics. Nevertheless, the successful deployment of portable sequencing for soil analysis validates the technical feasibility of field-based metagenomic workflows.

3.1.3 Wastewater-Based Epidemiology

Recent advances in wastewater-based epidemiology (WBE) have demonstrated the value of metagenomic surveillance for public health monitoring [13]. Digital PCR (dPCR) and droplet digital PCR (ddPCR) techniques have been successfully applied to detect emerging pathogens and track disease prevalence at the population level through municipal wastewater analysis. These applications underscore the growing acceptance of molecular diagnostics in non-clinical settings and validate the broader trend toward sequence-based pathogen identification. While WBE operates at a population surveillance scale rather than individual diagnostics, it shares with Pathogenius the challenge of analyzing complex microbial communities and extracting actionable information from samples containing DNA from multiple organisms.

3.1.4 Market Positioning

Pathogenius occupies a distinct position in the pathogen detection landscape by combining several attributes that existing solutions do not simultaneously provide: unbiased metagenomic analysis, operation on modest local hardware, offline functionality, and a user interface designed for non-specialists. Commercial PCR kits offer speed and simplicity but lack the discovery capability needed when pathogen identity is uncertain [7, 8]. Cloud-based metagenomic platforms provide comprehensive analysis but require reliable internet connectivity and raise data privacy concerns when handling clinical samples. Research tools like Nanometa Live deliver sophisticated analysis but assume access to computational infrastructure typically available only in laboratory settings.

The target market for Pathogenius includes emergency response teams, field hospitals, resource-limited clinics, and research expeditions where traditional laboratory infrastructure is unavailable or impractical. These scenarios prioritize rapid deployment, independence from external dependencies, and the ability to provide actionable information even under adverse conditions. The system's offline capability and modest hardware requirements directly address these operational constraints, while the species-level reporting format aligns with the information needs of clinicians and public health workers who require clear, interpretable results rather than exhaustive taxonomic profiles.

3.2 Academic Analysis

The academic literature provides substantial evidence supporting the technical feasibility and clinical utility of metagenomic pathogen detection while also revealing important challenges that Pathogenius must address.

3.2.1 Foundational Metagenomics Research

Metagenomic sequencing has fundamentally transformed microbiology by enabling culture-independent analysis of microbial communities [14]. Traditional culture-based methods fail to detect organisms that cannot be readily grown in laboratory conditions, creating blind spots in pathogen surveillance. Metagenomic approaches circumvent this limitation by directly

sequencing DNA extracted from clinical or environmental samples, providing an unbiased view of all organisms present. However, this comprehensiveness introduces computational and interpretive challenges: metagenomic datasets are inherently complex, containing sequences from host organisms, environmental contaminants, commensal microbiota, and potential pathogens in unknown proportions. Distinguishing clinically relevant findings from background noise requires sophisticated analysis methods and careful quality control.

Research has demonstrated that metagenomic analysis can successfully identify uncultivable microorganisms with potential applications in pharmaceutical and food industries [15]. These applications leverage the discovery aspect of metagenomics to characterize novel enzymes, biosynthetic pathways, and bioactive compounds from environmental samples. While Pathogenius focuses on pathogen identification rather than functional genomics, these studies validate the principle that sequence-based analysis can extract meaningful biological information from complex microbial mixtures without requiring pure cultures or prior knowledge of sample composition.

3.2.2 Next-Generation Sequencing Technologies

The evolution of next-generation sequencing (NGS) technologies has been crucial to making field-based metagenomics feasible [16]. Third-generation long-read sequencing platforms, particularly Oxford Nanopore Technologies' MinION and PromethION devices, offer several advantages relevant to Pathogenius: rapid sample preparation (4-6 hours), elimination of PCR amplification steps that can introduce bias, fast turnaround times with runs completing within a day, and read lengths substantially longer than second-generation sequencing technologies [17]. These long reads improve taxonomic classification accuracy by spanning larger genomic regions, enabling species-level discrimination even when organisms share similar sequence motifs.

The absence of PCR amplification in the library preparation workflow is particularly significant for clinical metagenomics. PCR-based methods can preferentially amplify certain DNA fragments over others due to differences in GC content, primer binding efficiency, or template structure, potentially distorting the quantitative representation of organisms in the sample. Direct sequencing approaches minimize this bias, providing a more accurate picture of

community composition. However, long-read technologies currently exhibit higher per-base error rates compared to short-read platforms, necessitating careful consideration of how classification algorithms handle sequencing errors.

3.2.3 Aquaculture and Environmental Monitoring

Recent research has explored metagenomic pathogen monitoring in sustainable freshwater aquaculture production, identifying opportunities to prevent disease outbreaks in fish farming through early detection of bacterial and parasitic pathogens [18]. These studies demonstrate that metagenomic surveillance can be effectively applied to non-human clinical samples and can operate in field settings where traditional laboratory infrastructure is limited. The parallels to human clinical diagnostics are instructive: both applications require rapid identification of pathogens from complex samples, both must distinguish pathogenic organisms from commensal or environmental microbiota, and both benefit from portable, cost-effective analysis platforms.

The aquaculture literature also reveals important limitations of metagenomic approaches that are equally relevant to human diagnostics. Quantitative accuracy remains challenging: the number of sequencing reads assigned to a particular organism does not necessarily correlate linearly with its abundance in the original sample due to differences in genome size, DNA extraction efficiency, and sequencing biases. Furthermore, detecting organisms present at very low abundance requires deeper sequencing coverage, extending analysis time and increasing computational demands. These constraints inform realistic expectations for Pathogenius: the system can identify the major constituents of a sample and detect abundant pathogens reliably, but may struggle with rare organisms or samples dominated by host DNA.

3.2.4 Computational Methods and Classification Algorithms

The choice of Kraken2 as the primary classification engine for Pathogenius is supported by extensive benchmarking studies demonstrating its favorable balance of speed, accuracy, and memory efficiency. K-mer-based classification methods operate by decomposing sequencing reads into short substrings of fixed length, then comparing these k-mers against a

pre-indexed reference database to identify the most likely taxonomic origin. This approach is computationally efficient because it avoids expensive sequence alignment operations, enabling classification of millions of reads on consumer grade hardware provided that the reference database is curated to fit within the available RAM.

However, k-mer methods have inherent limitations that must be understood when interpreting Pathogenius' results. Firstly, unlike alignment based methods, k-mer matching is sensitive to sequencing errors making robust preprocessing critical. Secondly, organisms sharing similar genomic sequences or sharing motifs via Horizontal Gene Transfer may be difficult to distinguish, particularly at the species level. In such cases Kraken2 gives lower resolution taxonomic labels. Finally, the accuracy of k-mer classification depends critically on the comprehensiveness and quality of the reference database: organisms not represented in the database cannot be identified, and closely related species may be misclassified if their distinguishing genomic features are not adequately captured.

3.2.5 Technical Validation and Deployment Considerations

Database curation represents a critical success factor. Reference databases must be regularly updated to include newly sequenced genomes and emerging pathogen variants [19] . They must also be carefully constructed to avoid taxonomic mislabeling or redundancy that could bias classification results. For Pathogenius, this implies an ongoing maintenance requirement beyond initial system development: as new pathogen genomes are published, the reference database should be updated to maintain classification accuracy.

3.2.6 Regulatory and Clinical Validation Requirements

The academic literature reveals a significant gap between technical proof-of-concept and clinically validated diagnostic tools. While numerous studies have demonstrated the technical feasibility of metagenomic pathogen detection [20], relatively few systems have undergone the rigorous clinical validation required for regulatory approval as diagnostic devices [21]. This validation process typically requires demonstrating analytical sensitivity (ability to detect low pathogen loads), analytical specificity (ability to distinguish target pathogens from

related organisms), and clinical sensitivity and specificity (agreement with reference diagnostic methods on patient samples).

Pathogenius is explicitly positioned as a decision support tool rather than a definitive diagnostic device, which partially addresses regulatory concerns by clearly communicating that results require clinical interpretation and should not be used as the sole basis for treatment decisions. However, this positioning must be maintained consistently throughout the user interface and documentation to ensure appropriate use. The academic literature on clinical decision support systems emphasizes the importance of transparent communication about system limitations, uncertainty quantification, and clear guidance on when human expert review is required.

4 Glossary

1. **FASTQ:**

   A text-based file format used to store raw sequencing reads along with per-base quality scores. Each entry contains a read identifier, the nucleotide sequence, a separator line, and a line encoding quality values. FASTQ files may include DNA from multiple organisms in unknown proportions.

2. **FASTA:**

   A sequence file format used to store reference genomes or assembled sequences. Each entry begins with a descriptor line preceded by “>”, followed by nucleotide sequences. Kraken2 reference databases typically consist of species-specific FAST files.

3. **Nanopare-Style Reads:**

   Long-read DNA sequences produced by Oxford Nanopare Technologies (ONT).

4. **Metagenomics:**

   A sequencing-based approach for analyzing the genetic material of all organisms present in a sample. Used in clinical or environmental diagnostics to detect pathogens without culturing.

5. **Kraken2:**

   A high-performance k-mer-based taxonomic classifier that assigns each read to the most likely species by comparing its short sequence fragments to a reference database. Used in Pathogenius for species-level pathogen identification.

6. **K-mer:**

   A substring of length k extracted from a DNA sequence. Kraken2 uses k-mers to rapidly compare read fragments to labeled genomic references.

7. **Snakemake:**

   A workflow management system that ensures reproducible execution of multi-step pipelines. It automatically handles dependencies, parallelization, and modular design of the Pathogenius workflow.

8. **Workflow:**

   A structured sequence of analysis steps executed automatically by Snakemake. Ensures deterministic and maintainable data processing.

9. **Reference Database:**

   A curated collection of genomic sequences in FASTA format used by Kraken2 to classify unknown reads. Each species has its own genome entry or entries, enabling species-level interface.

10. **Species-Level Identification:**

    Reporting only the organism species rather than full taxonomic hierarchy. This aligns with clinical needs and simplifies interpretation for non-experts.

11. **Taxonomic Classification:**

    The process of assigning sequencing reads to biological categories such as species, genus, or family based on sequence similarity.

12. **Long-Read Sequencing:**

    Sequencing technologies capable of generating reads thousands of bases long. While error-prone, long reads improve pathogen detection by covering larger genomic regions.

13. **Icarus Simulator:**

    A software tool used for testing Pathogenius in the absence of a physical Nanopare device.

14. **CUDA:**

    A parallel computing platform for GPU acceleration developed by NVIDIA.

15. **NCBI (National Center for Biotechnology Information):**

    A U.S. government-funded scientific organization that maintains some of the world's most widely used biological databases, including GenBank, RefSeq, and taxonomy resources. Many bioinformatics tools, such as Kraken2, rely on NCBI reference genomes and taxonomic identifiers for sequence classification.

16. **PCR:** Polymerase chain reaction.

5 References

[1] M. A. Schon *et al.*, “NanoPARE: Parallel analysis of RNA 5' ends from low-input RNA,” *Genome Research*, vol. 28, no. 12, pp. 1929–1937, Dec. 2018.

[2] Icarus-sim, “Icarus,” GitHub repository. [Online]. Available: https://github.com/icarus-sim/icarus. [Accessed: Nov. 26, 2025].

[3] OpenJS Foundation, “Electron,” [Online]. Available: https://www.electronjs.org/. [Accessed: Nov. 27, 2025].

[4] *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*, ISO/IEC 25010:2011, International Organization for Standardization, 2011.

[5] *Unified Modeling Language (UML) Specification*, Version 2.5.1, Object Management Group, Dec. 2017. [Online]. Available: https://www.omg.org/spec/UML/2.5.1.

[6] *Systems and software engineering — Life cycle processes — Requirements engineering*, ISO/IEC/IEEE 29148:2018, International Organization for Standardization, 2018.

[7] Bio-Rad Laboratories, “Pathogen Detection,” [Online]. Available: https://www.bio-rad.com/en-tr/a/ls/pathogen-detection. [Accessed: Nov. 27, 2025].

[8] Norgen Biotek Corp., “Waterborne Pathogen Detection,” [Online]. Available: https://norgenbiotek.com/category/waterborne-pathogen-detection. [Accessed: Nov. 27, 2025].

[9] G. Qiu, X. Zhang, and A. J. deMello, “On-site airborne pathogen detection for infection risk mitigation,” *Chem. Soc. Rev.*, vol. 52, no. 24, pp. 8531–8579, Dec. 2023. doi: 10.1039/D3CS00417A.

[10] FOI-Bioinformatics, “nanometa_live,” GitHub repository. [Online]. Available: https://github.com/FOI-Bioinformatics/nanometa_live. [Accessed: Nov. 25, 2025].

[11] L. Mak, B. Tierney, W. Wei, C. Ronkowski, R. B. Toscan, B. Turhan, M. Toomey, J. S. A. Martinez, C. Fu, A. G. Lucaci, A. H. B. Solano, J. C. Setubal, J. R. Henriksen, S. Zimmerman, M. Kopbayeva, A. Noyvert, Z. Iwan, S. Kar, N. Nakazawa, D. Meleshko, D. Horyslavets, V. Kantsypa, A. Frolova, A. Kahles, D. Danko, E. Elhaik, P. Labaj, S. Mangul, The International

MetaSUB Consortium, C. E. Mason, and I. Hajirasouliha, “CAMP: A modular metagenomics analysis system for integrated multi-step data exploration,” *bioRxiv*, 2023. doi: 10.1101/2023.04.09.536171.

[12] L. E. Braley, J. B. Jewell, J. Figueroa, J. L. Humann, D. Main, G. A. Mora-Romero, N. Moroz, J. W. Woodhall, R. A. White III, and K. Tanaka, “Nanopore Sequencing with GraphMap for Comprehensive Pathogen Detection in Potato Field Soil,” *Plant Disease*, vol. 107, no. 8, pp. 2288–2295, Aug. 2023.

[13] G. Jiang, R. Honda, and S. Arora, “Pathogen Detection and Identification in Wastewater,” *Water*, vol. 16, no. 4, p. 611, Feb. 2024. doi: 10.3390/w16040611.

[14] R. R. Miller, V. Montoya, J. L. Gardy, et al., “Metagenomics for pathogen detection in public health,” *Genome Med*, vol. 5, no. 81, p. 81, Dec. 2013. doi: 10.1186/gm485.

[15] L. M. Coughlan, P. D. Cotter, C. Hill, and A. Alvarez-Ordóñez, “Biotechnological applications of functional metagenomics in the food and pharmaceutical industries,” *Front Microbiol*, vol. 6, p. 672, Jun. 2015. doi: 10.3389/fmicb.2015.00672.

[16] H. Emteborg *et al.*, “Reference materials for the detection of genetically modified organisms,” Joint Research Centre (European Commission), Tech. Rep. JRC92395, 2014. [Online]. Available: https://publications.jrc.ec.europa.eu/repository/bitstream/JRC92395/lbna26881enn.pdf.

[17] J. K. Kulski, ‘Next-Generation Sequencing — An Overview of the History, Tools, and “Omic” Applications’, Next Generation Sequencing - Advances, Applications and Challenges. InTech, Jan. 14, 2016. doi: 10.5772/61964.

[18] D. B. Macedo, T. M. C. dos Anjos, E. F. F. De Los Santos, M. D. N. Rodrigues, O. V. Cardenas Alegria, and R. T. J. Ramos, “New perspectives on metagenomic analysis for pathogen monitoring in sustainable freshwater aquaculture production: a systematic review,” *Frontiers in Freshwater Science*, vol. 2, May 2024. doi: 10.3389/ffwsc.2024.1459233$.

[19] J. E. Allen, T. S. Brettin, and J. M. Kimbrel, “Addressing the dynamic nature of reference data: a new nucleotide database for robust metagenomic classification,” *mSystems*, vol. 10, no. 4, p. e01239-24, Apr. 2025.

[20] R. Schlaberg, C. Y. Chiu, S. Miller, G. W. Procop, and G. Weinstock, “Validation of Metagenomic Next-Generation Sequencing Tests for Universal Pathogen Detection,” *Arch Pathol Lab Med*, vol. 141, no. 6, pp. 776-786, Jun. 2017. doi: 10.5858/arpa.2016−0539−RA.

[21] B. Goldberg, H. Sichtig , C. Geyer , N. Ledeboer , G.M. Weinstock. Making the Leap from Research Laboratory to Clinic: Challenges and Opportunities for Next-Generation Sequencing in Infectious Disease Diagnostics. mBio. 2015 Dec 8;6(6):e01888-15. doi: 10.1128/mBio.01888-15. PMID: 26646014; PMCID: PMC4669390.