CS491 SENIOR DESIGN PROJECT

CASSIE

PROJECT SPECIFICATION REPORT

Ege Şirvan, Eren Aslan, Arda Kırcı, Osman Baktır, Emre Can Yoloğlu
Supervisor: Can Alkan

1. Introduction

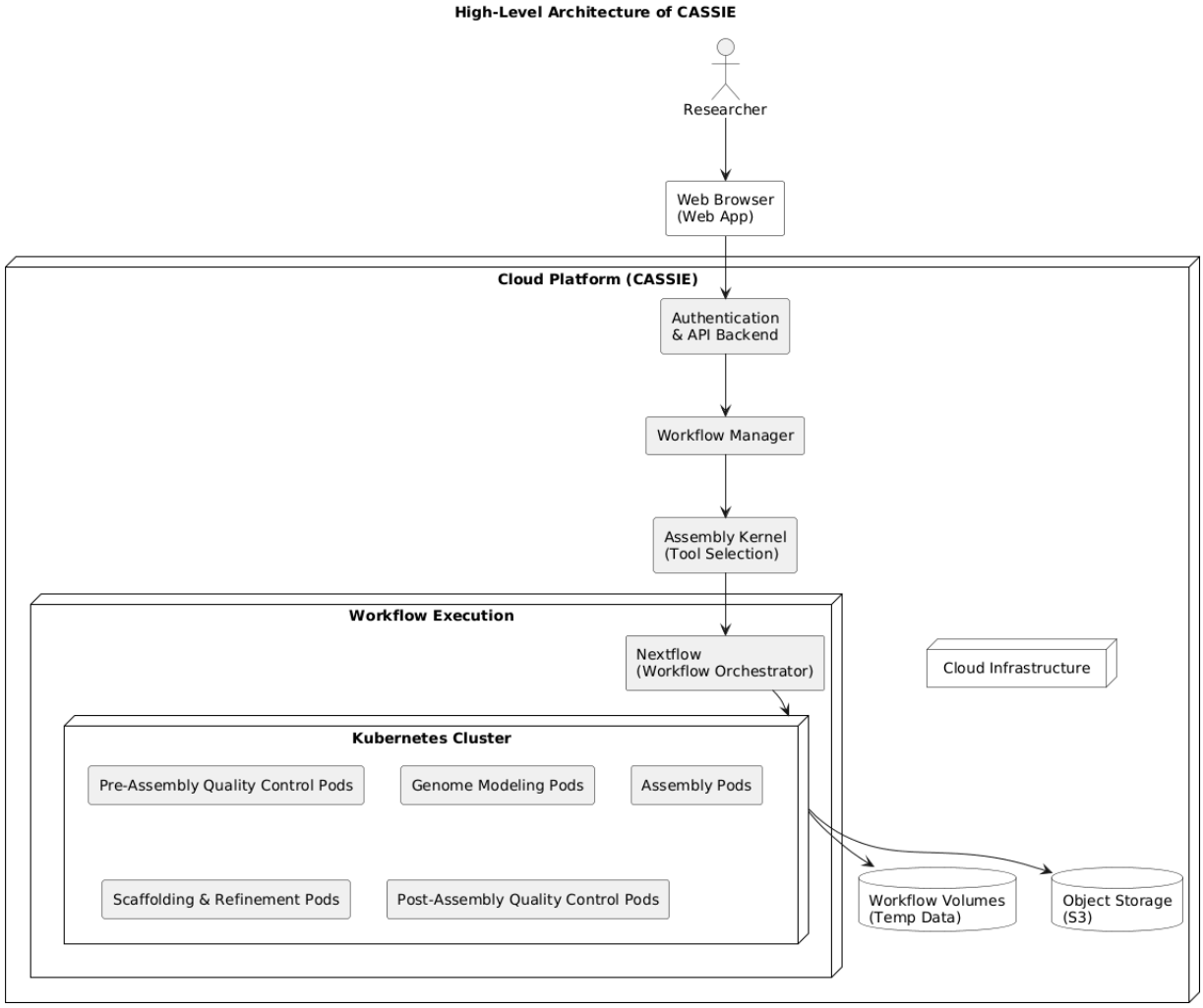
1.1. Description

CASSIE is a cloud-based genome assembly and annotation platform that aims to automate complex genome analysis processes and make large-scale genome assembly procedures more accessible to researchers. Modern workflows for genome assembly require multiple different tools to be integrated, highly compute-intensive steps to be managed, and knowledge about advanced bioinformatic topics. CASSIE eliminates this complexity by presenting a streamlined environment where users can upload their sequencing data, select appropriate tools, and run end-to-end assembly workflows from a single web interface. CASSIE addresses a wide range of research populations by supporting both human and nonhuman genomic data and provides powerful mechanisms for both data privacy and isolation.

CASSIE automates the fundamental steps of a standard genome assembly process: Quality control, estimation of genomic properties, assembly, and assembly quality assessment. CASSIE also integrates fundamental repeat annotations, and when appropriate, gene annotations. All tools have been containerized using Docker, and workflow conduction is managed by Nextflow. This architecture allows reproducibility, modularity, flexible scalability, and adaptability to different data types.

The infrastructure of CASSIE is designed to go with a scalable cloud architecture. During the development phase, MinIO and local Kubernetes distributions (Minikube) are being used, whereas the production environment is designed for a Kubernetes cluster that runs on AWS and has a VM pool that scales automatically. Kubernetes namespaces, resource quotas per workflow, and automated pod management make multi-tenant working safe and efficient. CASSIE helps researchers focus directly on scientific results instead of dealing with calculation infrastructure by abstracting these operational complexities completely from the end user.

1.2. High-Level Architecture and Components of Proposed Solution



1.3. Constraints

1.3.1. Implementation Constraints

The development and operation of CASSIE involve various technical and operational constraints. Genome assembly tools require high CPU and memory; some assemblers are specifically optimized for long read data (PacBio, HiFi, ONT). The system needs Docker-based containerization and Nextflow workflow management; thus, all the tools must be container-compatible with each other.

Since the local development environments run with limited resources on MinIo and Minikube/KIND, large datasets and full-scale assemblies cannot be tested without moving them to the production environment. In the production environment, the AWS-based Kubernetes cluster is used; this structure requires extra configurations such as IAM role management, namespace isolation, and an auto-scaling

VM pool. Since data transfer and storage processes involve working with large files, they pose additional constraints for bandwidth and latency.

1.3.2. Economic Constraints

There are direct economic constraints since CASSIE's production environment runs on the cloud. Costs may increase according to Kubernetes nodes that run on AWS, an auto-scaling VM pool, and S3 storage. Storage of large sequencing data creates a significant storage cost in the long term. Furthermore, additional costs arise associated with network usage when the users upload data to S3 and download the results. Given that the student team of the project has a limited budget, the majority of the tests are conducted in a local environment by emulating cloud systems, and cloud costs are minimized.

1.3.3. Ethical Constraints

Since the system might store human genomic data, it has significant ethical constraints. Human genome data is considered personal and sensitive, which is why ensuring data privacy, integrity, and restricting access solely to the user is mandatory. Because of the multi-tenant system architecture, it is necessary to completely isolate user data from each other.

Log systems should not contain any sensitive biological data and should retain only minimal data solely for technical debugging purposes. Under no circumstances can the genomic data that is used in the system be processed for non-research purposes, shared, or transmitted to third parties by CASSIE.

1.4. Professional and Ethical Issues

During CASSIE's design and development, professional responsibilities and ethical obligations play a crucial role. Genomic data, especially the human genome, is considered highly sensitive; thus, every part of the system must be designed with privacy, security, and data integrity principles. Within the framework of professional responsibility, user data must only belong to the owner, access control must be applied properly, and the data mustn't be shared with unauthorized third parties.

In scientific research, transparency and reproducibility are some of the core professional principles. For this reason, versions of the tools used in CASSIE, parameters, and workflow steps are logged by the system and presented to the user. In this way, verifiability and repeatability of the acquired results are secured.

Fairness in resource allocation is also a thing to consider as a professional issue in a multi-tenant system. In order to prevent users from consuming excessive resources and prevent other users' operations, resource quotas, prioritization, and isolation mechanisms are put in place within Kubernetes. This approach allows a user model that is both technically sustainable and ethically fair.

Additionally, software, libraries, and tools that are used to develop this system are integrated in compliance with their license terms. During the usage of open-source tools, copyright, license, and distribution conditions are considered, and all the components are referenced accordingly. This framework fits with both academic and professional ethical necessities.

1.5. Standards

During CASSIE's development, both software engineering standards and bioinformatic-specific data format standards must be complied with. These standards ensure sustainability, reproducibility, and compatibility with different platforms.

Software and Engineering Standards:

- **IEEE 830 – Software Requirements Specification Standard:**
  Principles of this standard are taken into account within the requirements section of the project documentation.
- **UML 2.5.1 – Modeling Standards:**
  UML guidelines are adopted for any diagrams utilized as needed (class diagrams, component diagrams, etc.).
- **OCI (Open Container Initiative) Standards:**
  Creation, portability, and execution of Docker containers are performed in compliance with OCI standards.
- **Kubernetes API ve Configuration Standards:**
  Component definitions such as Pod, Deployment, Namespace, and Resource quota are configured in compliance with official Kubernetes API conventions.
- **Nextflow DSL2 Standard:**
  Nextflow's DSL2 syntax is used to make sure the workflow is defined in a modular, reusable, and transparent structure.

Bioinformatic Data Format Standards:

Tools that CASSIE integrates use established and widely accepted data formats. Compliance with these formats ensures data flow within different tools without any problems.

- **FASTQ:** Stores raw character data produced by the sequencer along with confidence values for each character.
- **FASTA:** Stores raw character data produced by the sequencer.

- **GFA (Graphical Fragment Assembly):** A graph format where nodes represent sequence fragments and edges represent connections between them.
- **GFF3 / GTF:** Standard format used to store annotation data. Standard format used to store annotation data.
- **BAM / CRAM / SAM:** Log-like formats showing where each read maps onto a reference sequence; BAM/CRAM are compressed versions of SAM.
- **BED:** A simple tabular interval format specifying start–end positions of regions on a long sequence.

Compliance with these standards allows CASSIE to work in full compatibility with both modern cloud-based infrastructures and the existing tools and data structures within the bioinformatics ecosystem.

2. Design Requirements

2.1. Functional Requirements

Requirements below define the core functionality that the CASSIE system must fulfill. Requirements are delivered in a numbered, transparent, and traceable way.

User and Access Requirements

1. The system should allow users to log in via authentication.

2. Each job should be assigned a dedicated Kubernetes namespace where its processes and data are completely isolated from other jobs.

3. The data storage area (S3) for each user should be configured independently from other users.

Data Upload and Management Requirements

1. The system should allow users to upload sequencing files such as FASTQ/FASTA.

2. The system should automatically save uploaded data to the respective user's isolated storage area.

3. The system should validate the file formats of uploaded data and provide feedback to the user in case of incorrect formats.

4. The system should allow storing data from the user's own cloud storage to CASSIE’s storage without downloading data to their system.

Workflow Configuration Requirements

1. The user should be able to select the tools that they want to use.

2. Based on the tools selected, the system should show the necessary parameter spaces automatically.

3. The system should validate user input parameters and provide warnings for incorrect or missing parameters.

4. The system should allow the user to start multiple workflows with different configurations on the same dataset.

5. The system should ask high-level questions about jobs that the user wants to perform and decide on tools automatically.

6. The system should allow users to build their own workflows without automation.

7. The system should estimate required resources, expected time, and expected total price for all available EC2 instance classes, where each class has different available resources (e.g., total memory per user, total CPU per user).

8. The system should recommend an EC2 instance class according to the resource requirements and expected time.

9. The system should allow the user to select an EC2 instance class.

10. The system should execute the finalized workflow on the selected EC2 instance class.

Workflow Execution Requirements

1. The system should submit the user's workflow to Nextflow, which will execute it on Kubernetes using the Kubernetes executor.

2. Each Nextflow task belonging to a workflow step (QC, assembly, assessment, etc.) should run in its own Kubernetes pod.

3. Kubernetes should schedule pods according to the CPU and memory requests/limits configured for each pod and the resource quotas defined for the corresponding namespace.

4. The system should manage temporary files and intermediate outputs in dedicated storage (e.g., per-workflow volumes or buckets) associated with the namespace of the workflow, and clean them up when the workflow completes or fails.

5. Nextflow must be configured to automatically retry failed tasks (pods) according to a predefined retry policy when a pod fails during a workflow.

Post-Execution Requirements

1. The system should store the status (started, working, completed, error) information for each step and notify the user.

2. All desired results should be written to the user's storage directory.

3. After the workflow is completed, the user should be able to download the result files one by one or as a complete package (.zip/.tar.gz)

4. The system should output readable workflow logs to the user.

Tool Requirements

1. The system should be able to perform quality control analysis.

2. The system should be able to perform genome size, heterozygosity, and repeat content estimation.

3. The system should be able to perform genome assembly.

4. The system should be able to perform scaffolding, polishing, and gap filling.

5. The system should be able to perform assembly assessment.

6. The system should be able to perform repeat and segmental duğlication analysis.

7. The system should be able to perform gene annotation.

System Integration Requirements

1. In the production environment, the system should be integrated with AWS S3 and AWS EKS (Kubernetes).

2. The system should be capable of pulling container images from Docker Hub or a private image registry.

3. The system should guarantee both namespace and storage isolation for a multi-tenant architecture.

2.2. Non-Functional Requirements

This section defines the criteria that CASSIE must meet in terms of performance, usability, security, supportability, and scalability. These requirements include the core features that impact the system's user experience and operational stability.

2.2.1. Usability

1. The system should provide a clear and simple web interface that enables users to run genome assembly workflows step-by-step with ease.

2. The interface should only show the parameters for the related tools and reduce unnecessary complexity.

3. The error messages should be presented in a way that is readable, clear, and guides the user towards the correct solution.

4. Workflow status (ready, working, completed, error) should be output to the user via real-time or near-real-time updates.

5. The users should be able to access previous jobs, results, and logs from a single panel.

6. The users should be able to access and execute workflows from other users if they have shared them.

2.2.2. Reliability

1. The system should give readable error notifications in the event of a failure in the workflow steps.

2. User data shall be stored in a durable storage layer to prevent data loss in the event of network outages or system failures.

3. Critical system services shall run under Kubernetes controllers that automatically recreate pods if they terminate unexpectedly.

4. Error handling and retry mechanisms should be present when a connectivity error occurs between system components (ex., Nextflow → S3 access).

5. All workflows shall be designed to produce consistent and reproducible results when re-executed with the same parameters, tool versions, reference data versions, and random seeds.

2.2.3. Scalability

1. The system should support multiple workflows started by multiple users.

2. The Kubernetes cluster should provide horizontal scalability by adding new nodes when system load increases.

3. User-specific namespace architecture should guarantee resource isolation even under heavy load.

4. The storage system (S3) should be configured in a way that allows high volume and large numbers of concurrent read/write operations.

5. The system architecture should be designed with the flexibility to increase the number of tools if needed.

3. Feasibility Discussions

3.1. Market and Competitive Analysis

3.1.1 Target Users

CASSIE is targeted to scientists who work in the field of bioinformatics, especially scientists who work with genomic data but may lack technical expertise in genome assembly tools, command-line environments, or workflow orchestration. Additionally, it will also support bioinformatic scientists of all backgrounds since it decreases the effort and time to perform complex genome assemblies and annotations by streamlining multi-step processes, minimizing manual intervention, and accelerating large-scale analyses.

3.1.2 Market Research

The global genomics market, valued at USD 32.65 billion in 2023 and is projected to reach USD 94.86 billion by 2030 [1]. This expected increase is due to the increasing demand for personalized medicine, large-scale DNA sequencing projects (e.g, UK BioBank [2]), and the increasing number of sequencing platforms.

A key information in this market, which shows the potential of our project, is the fact that cloud-computing-based solutions have been steadily increasing their value and share of the market [1]. Another striking observation is based on the pricing model; the pay-as-you-go segment accounted for a market share of 55.0% in 2023, which is the pricing model CASSIE aims for [1].

Another major driver is the rise of long-read sequencing technologies (e.g., PacBio HiFi [3]), which produce significantly larger datasets than short-read technologies and require scalable compute environments. These technologies have accelerated demand for automated genome assembly and annotation platforms, particularly for large genomes or highly repetitive organisms. The long read sequencing market was valued at USD 538.9 Million in 2024, and is projected to reach USD 1.53 Billion by 2030, rising at a CAGR of 20.12% [4].

Researchers showed that even though this growth trend was realised in 2005, in 2015, it was concluded that there is a skilled-worker shortage [5]. In this growing market with a skilled-worker shortage, CASSIE provides a cloud-native platform that automates the entire workflow inside isolated and reproducible environments, which enables researchers with minimal computational expertise to perform complex genome analyses.

3.1.3 Competitive Analysis

Cloud-based genomic analysis platforms are well established in the industry, and most existing solutions follow similar design principles. Direct competitors include, but are not limited to, Galaxy, Terra Illumina. These generally provide a large number of analysis tools, workflow execution engines, access to public genomic datasets, and secure environments for storing sequencing data. Although these platforms are powerful, they usually expect users to have a strong computational background. They often require knowledge of workflow languages, containerization, and command-line interfaces. Many of these platforms are focused on short-read workflows or general-purpose analytics rather than providing streamlined, automated pipelines for complete genome assembly and annotation.

CASSIE differentiates itself from its competitors by focusing on end-to-end genome assembly and annotation, delivered through a fully automated and cloud-native environment. Every stage is handled through an intuitive interface that does not require command-line interaction or workflow engineering. By reducing both the operational burden and the expertise required to perform high-quality genome assemblies, CASSIE stands out as a specialized, user-centric platform that meets the evolving needs of modern genomics research.

3.2. Academic Analysis

3.2.1 Advances in Assembly Algorithms and Data-Driven Tool Selection

Hybrid long-read assembly strategies that combine HiFi and ultra-long ONT reads have produced near telomere-to-telomere assemblies in recent studies, such as the T2T Consortium [6], [7]. In light of T2T and similar studies, it is feasible to include state-of-the-art tools in the assembly systems.

Pre-assembly quality control [8] and k-mer spectrum modeling are widely recommended to characterize genome size, heterozygosity, and repeats before assembly, guiding correct tool selection and parameters [9]. This supports the inclusion of an “Assembly Kernel” that automatically selects the assembler [6], [10], [11] according to the input data profile.

3.2.2 Post-Assembly Refinement, Scaffolding, and Quality Assessment

Modern assembly pipelines rely on frameworks combining contiguity metrics [12], [13], gene completeness [14], [15], and reference-free k-mer validation [16], which together make up the field’s accepted standard for verifying structural and base-level accuracy [17]. Recent evaluations showed that assemblies using long-range data and systematic polishing show a better performance in continuity and correctness compared to pipelines that do not have refinement steps [9]. These findings justify adapting

automated refinement modules such as scaffolding, gap-closing, and duplication analysis within the platform.

3.2.3 Gene Annotation and Functional Characterization

Accurate genome analysis requires identifying genes, transcripts, and regulatory elements after assembly. Recent work shows that reliable results are possible by combining multiple sources of evidence, such as transcript data, known protein sequences, and repeat masks [18]. Modern tools [19], [20] can automatically predict genes, transfer known annotations, and assign basic biological functions. Including an automated gene annotation phase ensures that assembled genomes are not just structurally complete, but also biologically interpretable.

3.2.4 Reproducibility and Scalability via Cloud Automation

Genome assembly projects increasingly adopt containerized workflows and workflow-engine execution to ensure scalability and reproducibility across different datasets and computational environments, a trend highlighted in recent evaluations of cloud-native genomics pipelines [21], [22], [23]. Embedding automated quality control, consistent tool versions, and data-driven assembler selection allows platforms to deliver consistent, reproducible, and deterministic results at scale while reducing the computational expertise required from researchers.

4. References

[1] Grand View Research, “Genomics Market Size, Share & Trends Analysis Report,” 2024. Available: https://www.grandviewresearch.com/industry-analysis/genomics-market [Accessed Nov. 15, 2025].

[2] UK Biobank, “UK Biobank Research Resource,” 2024. Available: https://www.ukbiobank.ac.uk/ [Accessed Nov. 15, 2025].

[3] PacBio, “HiFi Sequencing Technology Overview,” 2024. Available: https://www.pacb.com/technology/hifi-sequencing/ [Accessed Nov. 15, 2025].

[4] Research and Markets, “Long-Read Sequencing Market: Size, Share, and Trends,” 2024. Available: https://www.researchandmarkets.com/reports/5649363/long-read-sequencing-market-size-share-and-trends#:~:text=The%20Long%20Read%20Sequencing%20Market,at%20a%20CAGR%20of%2020.12%25 [Accessed Nov. 15, 2025].

[5] S. Mulder et al., “A global landscape of bioinformatics training,” *Briefings in Bioinformatics*, vol. 20, no. 2, pp. 398-408, 2019. Available: https://academic.oup.com/bib/article/20/2/398/4096809 [Accessed Nov. 17, 2025].

[6] M. Rautiainen et al., “Telomere-to-telomere assembly of diploid chromosomes with Verkko,” Nature Biotechnology, 2023. Available (PMC Open Access): https://pmc.ncbi.nlm.nih.gov/articles/PMC10427740/ [Accessed Nov. 17, 2025].

[7] S. Nurk et al., “The complete sequence of a human genome,” Science, vol. 376, no. 6588, pp. 44-53, 2022, doi: https://doi.org/10.1126/science.abj6987 [Accessed Nov. 17, 2025].

[8] S. Andrews, "FastQC: A Quality Control Tool for High Throughput Sequence Data," Babraham Bioinformatics, 2010. Available: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/ [Accessed Nov. 27, 2025].

[9] E. Espinosa et al., "Comparing assembly strategies for third-generation sequencing data," article hosted on ScienceDirect, Elsevier, 2023. Available: https://www.sciencedirect.com/science/article/pii/S0888754323001441 [Accessed Nov. 17, 2025].

[10] Concepcion et al., "Haplotype-resolved assembler for accurate long reads," *Nature Methods*, 2021. Available: https://www.nature.com/articles/s41592-020-01056-5 [Accessed Nov. 27, 2025].

[11] A. Bankevich et *al.*, "SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing," *Journal of Computational Biology*, 2012. Available (PMC): https://pmc.ncbi.nlm.nih.gov/articles/PMC3342519/ [Accessed Nov. 27, 2025].

[12] A. Gurevich *et al.*, "QUAST: quality assessment tool for genome assemblies," *Bioinformatics*, 2013. Available: https://doi.org/10.1093/bioinformatics/btt086 [Accessed Nov. 27, 2025].

[13] H. Išerić et al., "Fast characterization of segmental duplication structure in multiple genome assemblies," *Algorithms Mol Biol*, vol 17, 2022. Available: https://doi.org/10.1186/s13015-022-00210-2 [Accessed Nov. 27, 2025].

[14] F. Tegenfeldt et al., "OrthoDB and BUSCO update: annotation of orthologs with wider sampling of genomes", *Nucleic Acids Research*, vol. 53, 2025, pp. 516-522. Available: https://doi.org/10.1093/nar/gkae987 [Accessed Nov. 27, 2025].

[15] A. Rhie, "Mercury: reference-free quality, completeness, and phasing assessment for genome assemblies," *Genome Biology*, vol. 21, no. 245, 2020. Available: https://genomebiology.biomedcentral.com/articles/10.1186/s13059-020-02134-9 [Accessed Nov. 27, 2025].

[16] Ranallo-Benavidez et al., "GenomeScope 2.0 and Smudgeplot for reference-free profiling of polyploid genomes," *Nature Communications*, vol. 11, no. 1432, 2020. Available: https://www.nature.com/articles/s41467-020-14998-3 [Accessed Nov. 27, 2025].

[17] Australian BioCommons, "Assess the quality of your genome assembly," *How-to Guides: Genome Assembly*, Australian BioCommons, 202Comparing assembly strategies for third-generation sequencing technologies across different genomes - ScienceDirect3. Available: https://australianbiocommons.github.io/how-to-guides/genome_assembly/assembly_qc [Accessed Nov. 17, 2025].

[18] R. Hubley et al., "RepeatMasker Open-4.0," Institute for Systems Biology, 2013. Available: https://www.repeatmasker.org/ [Accessed Nov. 27, 2025].

[19] F.A.B. von Meijenfeldt *et al.*, "Comparative Annotation Toolkit (CAT)—simultaneous clade and personal genome annotation," *Genome Research*, 2018. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC6028123/ [Accessed Nov. 27, 2025].

[20] A. Shumate and S. Salzberg, "Liftoff: accurate gene annotation mapping across species," *Bioinformatics*, 2021. Available: https://doi.org/10.1093/bioinformatics/btaa1016 [Accessed Nov. 27, 2025].

[21] A. Rhie et al., "WebQUAST: online evaluation of genome assemblies," 2023. Article available via PMC: https://pmc.ncbi.nlm.nih.gov/articles/PMC10320133/ [Accessed Nov. 17, 2025].

[22] Y. Zhang et al., "GAEP: a comprehensive genome assembly evaluating pipeline," *Journal of Genetics and Genomics*, 2023. Available: https://www.sciencedirect.com/science/article/pii/S1673852723001194 [Accessed Nov. 17, 2025].

[23] The Galaxy Community, The Galaxy platform for accessible, reproducible, and collaborative data analyses: 2024 update, *Nucleic Acids Research*, vol. 52, pp. 83-84, 2024, https://doi.org/10.1093/nar/gkae410 [Accessed Nov. 17, 2025].