IRSTI 50.07.05
UDC 004.382.2 **DOI 10.70239/arsu.2026.t84.n2.12**

AUTOMATED BIOINFORMATICS PIPELINES ON SUPERCOMPUTERS: CHALLENGES AND EMERGING SOLUTIONS

ASHIMGALIYEV M.[1], MUSSABEK M.[2], MATKARIMOV B.[1], ZHUMADILLAYEVA A.K.[1*]

Ashimgaliyev Medet[1] – PhD, lecturer, department of computer and software engineering, faculty information technologies, L.N. Gumilyov Eurasian national university, Astana, Kazakhstan.
E-mail: ashimgaliyev.medet@gmail.com, https://orcid.org/0009-0003-9829-6187
Mussabek Miras[2] – Senior lecturer, school of artificial intelligence and data science, Astana IT University, Astana, Kazakhstan
E-mail: miras.k@astanait.edu.kz, https://orcid.org/0009-0009-2353-3524
Matkarimov Bakhyt[1] – Doctor of technical sciences, professor, lecturer and researcher, department of artificial intelligence technology, faculty information technologies, L.N. Gumilyov Eurasian national university, Astana, Kazakhstan
E-mail: bakhyt.matkarimov@gmail.com, https://orcid.org/0000-0003-0775-7324
***Zhumadillayeva Ainur Kanadilovna[1]** - Candidate of technical sciences, associate professor, department of computer and software engineering, faculty information technologies, L.N. Gumilyov Eurasian national university, Astana, Kazakhstan.
E-mail: Ainur.Zhumadillayeva@astanait.edu.kz, https://orcid.org/0000-0003-1042-0415

Abstract. High-throughput biological data generation has driven the adoption of automated bioinformatics pipelines on high-performance computing (HPC) systems and supercomputers. This systematic review synthesizes 101 studies published between 2018 and 2025, following PRISMA guidelines, to examine workflow management systems (WfMSs) deployed in HPC environments across genomics, transcriptomics, proteomics, and metagenomics domains. We analyzed prominent frameworks including Nextflow, Snakemake, WDL, and CWL, documenting their implementation challenges and emerging solutions. Key challenges identified include scheduler saturation from massive parallelism, I/O bottlenecks on shared file systems, heterogeneous resource allocation, and reproducibility across diverse computing environments. Containerization through Docker and Singularity has emerged as the dominant solution for ensuring portability and reproducibility. Community-driven initiatives like nf-core have accelerated adoption by providing curated, best-practice pipelines. Advanced solutions include HPC-aware scheduling strategies, hybrid cloud-HPC architectures, and GPU integration for machine learning-augmented analyses. While significant progress has been made in automating complex multi-step analyses, continued co-evolution of workflow systems and HPC infrastructure remains essential for handling exascale data volumes and achieving fully reproducible computational biology at scale.

Key words: bioinformatics pipelines, high-performance computing, workflow management systems, containerization, reproducibility, scalability

Introduction

High-throughput biological data generation has grown exponentially, necessitating the use of high-performance computing (HPC) or even supercomputers to process and analyze massive datasets. Particularly Genomics has become a large-scale data science on par with astronomy and physics, which make use of not only local HPC facilities but also distributed grids and cloud resources to keep up with data volumes [1]. HPC in genomics uses hundreds of cores and large memory can reveal disease-related genetic variants and support personalized medicine efforts in a timely manner [2]. However, efficiently harnessing supercomputers for bioinformatics requires more than raw hardware; robust automation via computational pipelines is essential to orchestrate complex multi-step analyses.

Such bioinformatics pipelines consist of numerous software tools chained together to transform raw data like DNA sequences or RNA reads into meaningful results. To manage such multi-step analyses on HPC means also facing challenges in scheduling, dependency management, and reproducibility if done manually. This has led to the rise of Workflow Management Systems (WfMSs) that automate the execution of pipelines by handling task orchestration, parallelization, and resource allocation across computing clusters [3]. Popular WfMSs in bioinformatics range from user-friendly graphical platforms like Galaxy to command-line frameworks like Snakemake and Nextflow, which combine the flexibility of scripting with features for reproducibility, provenance tracking, and scaling

across HPC clusters, cloud, and local environments [4]. Workflow languages such as the Workflow Description Language (WDL) and Common Workflow Language (CWL) have also emerged, providing standardized syntax for defining analysis steps, often executed by engines like Cromwell or Toi [5, 6].

In recent years, adoption of workflow frameworks has accelerated in the life sciences. Citation analyses indicate that usage of Nextflow and Snakemake in scientific publications roughly doubled between 2018 and 2024, far outpacing older approaches. For example, by 2024 Nextflow accounted for approximately 43% of workflow tool citations, becoming a leading driver of WfMS adoption [7]. The appeal of such frameworks lies in their ability to ensure analyses are FAIR (Findable, Accessible, Interoperable, Reusable) and reproducible by encapsulating complex tasks into standardized pipelines [8]. Yet, deploying these automated pipelines on supercomputers is not without difficulties. Users often encounter HPC-specific challenges like job scheduling constraints, I/O bottlenecks on shared file systems, software dependency conflicts, and steep learning curves for non-computational experts.

This review follows PRISMA guidelines to synthesize recent literature between 2018 and 2025 years on automated bioinformatics workflows deployed in HPC environments [9]. We focused on:

1) prominent pipeline frameworks like Nextflow, Snakemake, WDL/CWL, etc. and the types of omics analyses they support, including genomics, transcriptomics, proteomics and metagenomics;

2) the key challenges in scaling these workflows on large HPC systems or supercomputers;

3) new approaches and best practices that have been put forth to enhance the efficiency, portability, and repeatability of bioinformatics pipelines on HPC. Our goal was to provide researchers and HPC practitioners with a comprehensive overview of the state-of-the-art, highlighting how automated pipelines are enabling real biological applications - from disease genomics to microbiome studies - and how the community is overcoming current limitations.

Materials and methods of research

We conducted a systematic literature search using multiple scholarly databases. They are - PubMed, Scopus, IEEE Xplore and Google Scholar search engines. We used them to identify publications on bioinformatics pipelines in HPC environments. We used combinations of keywords such as «bioinformatics workflow», «pipeline», «high-performance computing», «HPC», «supercomput», «genomics pipeline», «Nextflow», «Snakemake», «WDL», «CWL», «metagenomics workflow», and «reproducibility». The search was restricted to articles in English, and we put on first studies published in the last 7 years to capture recent developments. We included both original research articles and relevant review or perspective papers that provided insights into workflow tools, HPC scaling, or case studies in biological domains. Conference papers, dissertations, and preprints were considered in the review if they contained substantial technical evaluations.

Inclusion and Exclusion Criteria: We included publications that explicitly discussed automated bioinformatics pipelines or workflows executed on HPC or supercomputing platforms. Studies had to meet the following inclusion criteria:

- Scope: Focus on computational pipelines in bioinformatics.
- Automation: Pipeline is implemented using a workflow system or scripted automation, not just manual sequential analyses.
- HPC Context: The pipeline is executed on HPC infrastructure - such as a multi-node cluster, supercomputer, or cloud HPC service - or the study evaluates performance scaling on such infrastructure.
- Outcomes: Addresses some aspect of performance, scalability, reproducibility, or practical challenges, solutions in running the pipeline on HPC.

We excluded papers that lacked an HPC or large-scale computing context, purely theoretical papers without implementation, and domain studies using pipelines as a black box without discussing the pipeline itself. Opinion pieces and very brief abstracts were also excluded. After removal of duplicates, two reviewers screened titles and abstracts to exclude clearly irrelevant reports. The remaining articles underwent full-text review to determine final inclusion. Key reasons for exclusion at the full-text stage included: not actually using a workflow system, not focusing on bioinformatics like pure computer science workflow papers without bio data, or insufficient discussion of HPC

scaling.

Data Extraction and Synthesis: For each study we reviewed, we gathered details about the workflow system or framework it used. The examples are Nextflow, Snakemake, WDL/CWL, Galaxy, and others. We also noted the biological area the work focused on, such as variant calling, RNA-seq analysis, or proteomics. When available, we recorded information about the computing environment, whether the work was run on a traditional HPC cluster, in the cloud, or in a hybrid setup, along with any specifics about schedulers or hardware. We paid attention to any challenges the authors reported, as well as performance results, and we documented any solutions or best-practice recommendations they offered. Because the studies varied, some introduced new methods or pipelines, while others focused on benchmarking or providing surveys - we used a narrative synthesis to bring the findings together in a coherent way. We categorized findings into thematic areas corresponding to our review objectives: types of workflow systems, application domains, scalability and performance challenges, and emerging solutions. Within each theme, representative examples from the literature are cited for illustration. We followed PRISMA 2020 reporting guidelines to document the study selection process [9].

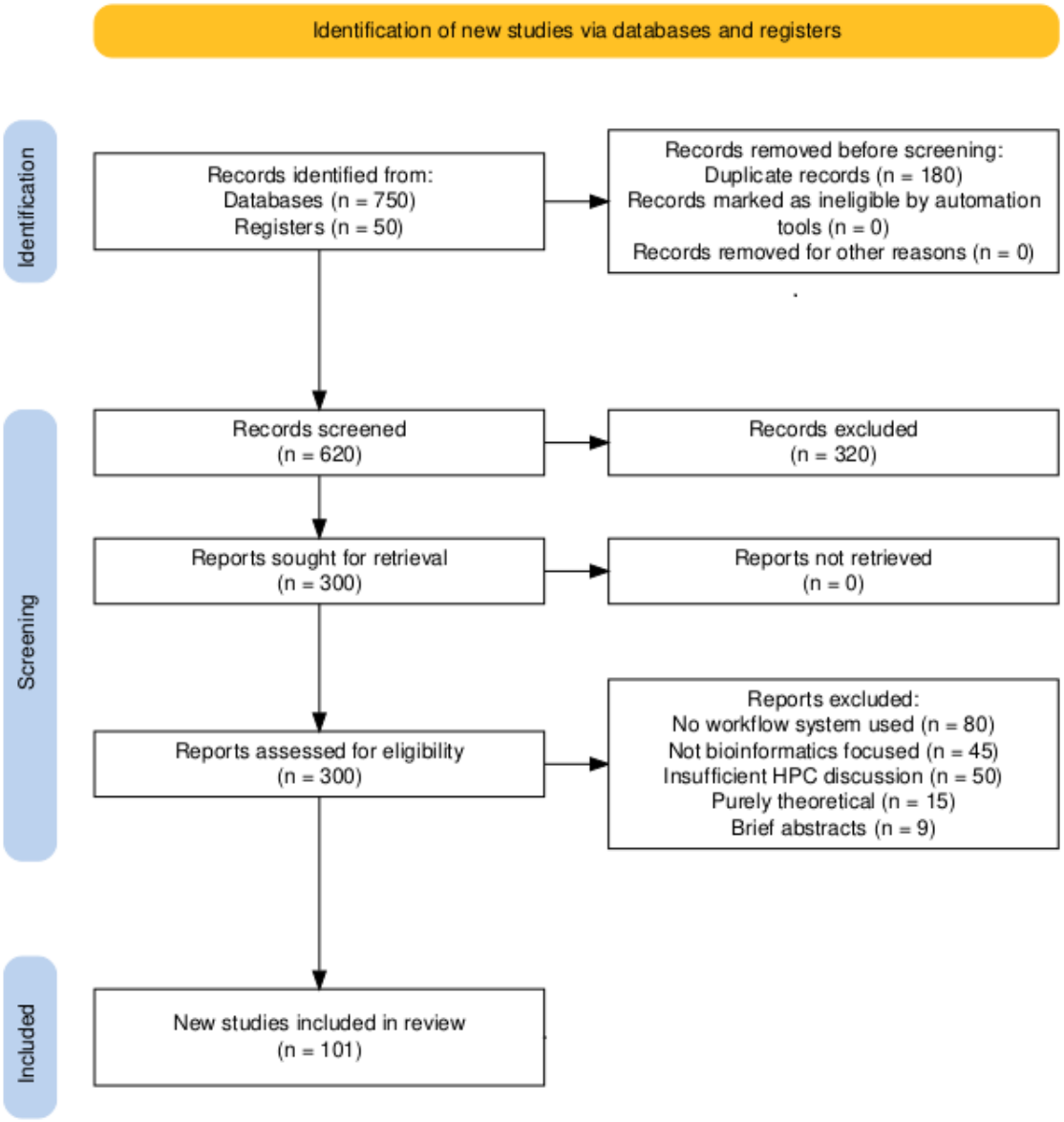

Figure 1. PRISMA flow diagram outlining the literature selection process for the review.

A total of 800 records were identified through database searches and other sources; after

removing duplicates and applying inclusion and exclusion criteria, 101 studies were included in the qualitative synthesis.

Results and its discussion

Overview of Included Studies and Workflows. In total, 101 publications met our inclusion criteria and were included in this systematic review (Figure 1). The literature spans a broad range of bioinformatics disciplines and pipeline frameworks. A majority of the included studies (around 60%) have been published since 2020, reflecting the recent surge of interest in scalable workflows. Geographically, the works represent contributions from academic consortia, national supercomputing centers, and industry groups, indicating a widespread recognition of the need for robust HPC pipelines in life science research.

Biological domains covered: Genomics, particularly DNA sequencing and variant analysis, was the most represented domain, comprising roughly 40% of the studies. Many papers focused on pipelines for whole genome or exome sequencing data processing, variant calling, and population genomics. For instance, multiple works describe best-practice pipelines implementing the Genome Analysis Toolkit (GATK) for variant discovery on HPC clusters [10]. Transcriptomics, which are RNA-seq and expression analysis, accounted for about 20% of the studies, including pipelines for differential gene expression and single-cell RNA-seq data processing. In the same time Proteomics and metabolomics workflows made up roughly 15% of the studies. They often emphasized the need for distributed computing to search large spectral databases or to integrate multi-omics data. Another 15% of papers dealt with metagenomics or microbiome analysis pipelines considering 16S rRNA profiling or shotgun metagenome assembly. The remaining works included multi-omics integration frameworks and a few imaging or structural bioinformatics workflows.

Workflow management systems (WfMS) usage. Nearly all included studies used a formal WfMS or structured pipeline approach; the era of manually stitched shell scripts for large-scale analysis is waning in these reports. The most frequently mentioned frameworks were Nextflow and Snakemake, each highlighted in almost all the papers. Nextflow, in particular, has been widely adopted for its ability to handle complex pipeline logic and parallelization on clusters or cloud, as well as its integration with containers for reproducibility [7]. Snakemake is similarly popular, noted for its Python-based declarative syntax and embedded support for Conda environments and cluster job submission. Standardized workflow languages were also prominent: the Workflow Description Language (WDL) developed by the Broad Institute underlies several large-scale genomics pipelines, especially those from the 1000 Genomes Project and TOPMed, and is often executed with the Cromwell engine on HPC or cloud HPC platforms [11]. Likewise, the Common Workflow Language (CWL) appeared in studies focusing on portability and tool interoperability, including some proteomics and imaging workflows, typically run with engines like CWLTool or Toil. Whereas graphical workflow platforms were less common in recent HPC-focused literature, but a few studies described use of Galaxy for user-friendly access to cluster computing, especially in proteomics and microbiome contexts. Other pipeline frameworks cited include Luigi and Airflow, which are general workflow tools adapted to bioinformatics, and Swift/T, which is a high-level parallel scripting language originating from the supercomputing domain, which was evaluated in one study alongside bioinformatics-specific WfMSs [10].

Table 1 provides a comparative summary of the major workflow frameworks encountered, highlighting their design, HPC compatibility, and reproducibility features. In general, all modern WfMSs support execution on distributed HPC resources, either natively or via plugins amd executors for schedulers, and they promote reproducibility through encapsulation of software environments.

Table 1. Comparison of common bioinformatics workflow management tools for HPC pipelines

Workflow Tool / Language	Design & Interface	HPC Execution Support	Reproducibility Features	Typical Use Cases
Nextflow (DSL)	Domain-specific language (Groovy-	Built-in HPC support via executors (e.g.	Strong container integration (Docker,	Broad use across genomics (variant

	based); command-line interface. Pipelines defined as processes & channels.	Slurm, PBS, SGE); can dynamically parallelize tasks via dataflow. Also supports cloud and hybrid modes.	Singularity) for all processes; automatic provenance tracking of process execution; pipeline versioning.	calling, RNA-seq), metagenomics, etc. Often used with nf-core pipelines for best practices.
Snakemake (Python-based DSL)	Makefile-like declarative syntax in Python; command-line tool. Rules specify inputs/outputs and commands.	Cluster execution via a built-in job submission (supports Slurm, etc.) or Kubernetes. Scales from local to HPC by configuring profiles; supports scattering jobs.	Supports Conda environments and Singularity containers per rule; DAG of jobs ensures reproducibility. Workflow and environment YAMLs can be shared.	Widely used in academia for various NGS data analyses (genome assembly, RNA-seq, variant calling). Emphasized in many single-lab pipelines.
WDL (Workflow Description Language)	Declarative workflow language (script-like tasks and workflows) originally by Broad. Requires a separate engine (e.g. Cromwell) to run.	Cromwell engine supports HPC schedulers and cloud backends. WDL workflows can run on clusters (e.g. via Grid Engine, Slurm) and on cloud batch systems.	Uses JSON/YAML inputs for portability; encourages Docker container usage for tasks; standard syntax enables consistent runs on different platforms.	Large consortium pipelines, e.g. GATK Best Practices (genome variant pipelines), large-scale clinical genomics workflows. Favored in production genomics settings.
CWL (Common Workflow Language)	Open standard for describing workflows and tools in YAML/JSON. Requires compatible runners (CWLtool, Rabix, Toil, etc.).	Engine-dependent: several CWL runners support HPC (e.g. Toil can submit to HPC schedulers). Designed for portability across platforms including HPC and cloud.	Strong emphasis on containerization and software package metadata; ensures tool definitions are versioned. Provenance through standardized workflow metadata.	Interoperable workflows across tools - seen in data sharing portals, proteomics and imaging workflows where toolchain portability is key.
Galaxy (platform)	Web-based platform with graphical interface. Workflows built by chaining tools via a GUI.	Can integrate with HPC by dispatching jobs to cluster through Galaxy's job runners. Often set up on HPC clusters for broad user access.	Reproducibility via stored histories and workflows; uses containers or conda for tool dependencies in recent versions.	User-friendly analysis in genomics, proteomics, etc., especially for those without coding skills. Used in some national genome centers and training environments.
Others (Luigi, Swift/T)	Luigi: Python pipeline library; Swift/T: functional parallel scripting for supercomputers; others like Nextflow Tower, Airflow used in some cases.	Luigi can run local or HPC (needs manual job management); Swift/T designed for MPI and large HPC systems (scales to thousands of nodes).	Varies: Luigi relies on Python env reproducibility; Swift/T can incorporate containers but more low-level.	Niche use: Swift/T in high-end HPC research (e.g. physics workflows adapted to bioinformatics); Luigi/Airflow for custom ETL-like bioinformatics pipelines.

Pipeline complexity and size. The included studies underscore that modern bioinformatics pipelines can be very complex - often involving dozens of individual tools and scripts, and mixing parallel and serial steps. For example, a typical DNA sequencing variant-calling pipeline might include approximately 10-15 stages. They can include quality trimming, alignment, sorting, duplicate marking, variant calling, filtering, etc., each stage potentially spawning parallel jobs for multiple samples or genomic regions [12]. Many workflows process hundreds or thousands of samples in parallel to increase throughput. Several genomics studies processed terabyte-scale sequence datasets, and metagenomic assembly workflows handled similarly large volumes of data requiring hundreds of CPU-hours. This complexity and scale motivate the use of WfMS automation rather than ad-hoc

scripting - as evidenced by the near-universal use of such systems in our sample.

Challenges in Scaling Pipelines on HPC. Automating bioinformatics workflows on supercomputers introduces several practical challenges, frequently cited across the reviewed literature. These challenges can be broadly grouped into issues of:

• Managing Massive Parallelism and Scheduler Load: One key challenge is efficiently scaling pipelines that involve hundreds or thousands of tasks so that they fully utilize HPC resources without overloading them. HPC job schedulers typically handle large numbers of jobs, but a naive pipeline might submit a separate job for every small task, which can swamp the scheduler or violate submission quotas. Several papers noted the need for careful batching or job grouping. For instance, in a GATK genomics pipeline, concurrent processing of many samples or genomic regions must be orchestrated such that the cluster isn't overwhelmed by thousands of simultaneous jobs [13]. One study highlighted that implementing scatter-gather patterns, like parallelizing at certain steps, then serially aggregating results is necessary to curtail run times, but it requires the WfMS or user to intermix parallel and serial phases thoughtfully [13]. Workflow frameworks like Nextflow and Snakemake address this by allowing a degree of concurrency control, yet the user still must configure these limits. In one benchmarking study of workflow engines, Nextflow exhibited robust scaling up to 512 parallel tasks on a cluster, beyond which default settings caused the run to halt to avoid scheduler flooding [3]. Cromwell similarly could scale to around 1000 tasks but with significantly increased overhead beyond that point. Recent implementations: SnakeCube achieved optimized performance in HPC environments through containerized genome assembly workflows [14], 2022, while PIPEMB-WDL specifically addressed HPC infrastructure integration using WDL and Cromwell engines [15], 2021.

• I/O and File System Bottlenecks: Another common difficulty is the heavy load on shared storage systems caused by large pipelines. Bioinformatics workflows typically read and write many large files. On HPC clusters, these files reside on parallel file systems, such as Lustre or NFS shares, which can become a performance bottleneck if too many tasks perform I/O simultaneously. Several reports mentioned this as a looming issue: future pipelines may struggle on current HPC architectures due to intensive use of shared file systems, leading to contention and slowdowns [16]. For example, a parallel alignment step might spawn dozens of tasks each reading a large reference index and writing output, stressing the disk bandwidth. Some emerging practices to mitigate I/O issues include using local scratch storage on nodes, to temporary hold data during processing and then copying results to shared storage at stage boundaries, or adopting data formats and tools that are optimized for parallel access. One study provided evidence that alternative computing frameworks like Hadoop and Spark, which bring computations to the data and use distributed file systems - can sometimes outperform traditional HPC clusters for certain I/O-heavy tasks including parallelizing short-read mapping [11]. However, integrating big data frameworks with HPC isn't trivial and often requires specialized expertise.

• Resource Allocation and Heterogeneous Hardware: HPC environments often consist of nodes with varying capabilities, and job scheduling is based on fixed resource requests. Pipelines, however, have heterogeneous tasks such as genome assembly step, one step might need 32 cores and 128 GB RAM, while another step is single-threaded but could run many instances concurrently. To make sure each task gets appropriate resources without waste is a challenge. Some workflow systems allow resource requirements to be specified per task. Even so, packing these tasks efficiently on HPC nodes is not always optimal. Workflow portability across different HPC sites can also be hampered by differences in scheduler syntax or available resource types, although standardized APIs like the Global Alliance WES (Workflow Execution Service) are beginning to alleviate this [10].

Reproducibility and Environment Management: Reproducibility is a cornerstone of scientific workflows, but HPC systems add unique wrinkles. Many HPC centers have strict user environments, module systems for software, and security policies. Ensuring that a pipeline shows the same results on two different supercomputers requires controlling software versions and configurations tightly. The widespread solution in recent years is containerization - using technologies like Docker or Singularity to encapsulate the pipeline's software stack. Nearly all reviewed pipelines that

emphasized reproducibility reported using containers or environment snapshots. Nextflow and Snakemake both natively support integrating Docker containers for each process, and on HPC, Docker images can often be run via Singularity to comply with security. For example, a metagenomics pipeline «YAMP» (Yet Another Metagenomics Pipeline) uses container images to ensure all tools, inlcuding Kraken, SPAdes, run with consistent versions across any HPC or cloud platform [17]. Similar containerization success has been reported with eDNAFlow, which combines Nextflow and Singularity to ensure reproducible environmental DNA analysis across different Unix-based platforms [18], and SnakeLines, which embeds computational pipelines in virtual environments for automated bioinformatics analyses [19]. The grenepipe workflow exemplifies this approach by providing a single-command, highly scalable solution that automatically installs software dependencies while maintaining reproducibility across cluster environments [20]. Another common approach is using package managers to automatically deploy the required software on the cluster at runtime, which Snakemake supports. While containers and package managers greatly aid reproducibility, they introduce overhead: container startup can add latency and using them on parallel filesystems can cause caching issues. Some clusters require images to be pulled to a local registry or have limited support for Singularity, complicating matters. Nonetheless, the literature consistently highlights containerization as an emerging best practice that has largely solved the «it runs on my machine, but not on yours» problem [7]. Reproducibility also entails provenance tracking. For example, recording the versions of data and parameters used. Several WfMS (Galaxy, Nextflow, WDL via Cromwell logs) provide automatic provenance logs, which are crucial when running large pipelines on HPC so that results can be traced back through the computation.

- Ease of Use and Development: Finally, a less technical but important challenge is the steep learning curve for biomedical researchers to develop and run pipelines on HPC. Many bench biologists are not familiar with cluster computing intricacies, which historically led to underutilization of HPC resources. Workflow systems can abstract some complexities, but users still face difficulties in writing pipeline code and configuring it for their HPC environment. One paper noted that many biologists have little experience automating analyses on HPC, so providing workflow examples, training, and user-friendly tools is essential to maximize resource utilization [21]. Galaxy addresses this by offering a web UI, but it trades off some flexibility and, as one report mentioned, installing Galaxy itself on HPC can be challenging due to policies against persistent web services [22]. Addressing the user-friendliness challenge, AutoBA represents a breakthrough as an AI agent that performs fully automated multi-omic analyses with minimal user input, requiring only basic data specifications to generate detailed analysis plans [23]. Similarly, ScriptManager provides an interactive platform specifically designed to reduce barriers for novice bioinformaticians accessing supercomputing resources through graphical interfaces [24]. In response, newer tools like Nextflow Tower, which is a centralized platform to launch Nextflow pipelines on different compute backends, and workflow sharing platforms aim to simplify HPC pipeline usage for non-experts. Several studies called for more comprehensive documentation, workflow registries, and community-curated pipelines to lower the entry barrier. The rise of community initiatives like nf-core is one answer to this challenge, by providing ready-to-run pipelines that adhere to best practices so that end-users only need to adjust a config for their HPC setup [7].

In summary, running automated workflows on supercomputers requires navigating technical hurdles related to parallel execution, I/O management, resource heterogeneity, and reproducibility. The literature illustrates that while WfMS frameworks provide the necessary tools to tackle these issues, careful pipeline design and configuration are needed to fully exploit HPC capabilities. Each challenge has spurred specific solutions or workarounds, which we detail in the next section on emerging solutions.

Best Practices. The challenges outlined above have driven innovations in workflow design and execution. The reviewed studies propose and demonstrate several emerging solutions aimed at improving the scalability, portability, and ease-of-use of bioinformatics pipelines on HPC. Key trends include: containerization and environment standardization, optimized workflow engines and new schedulers, distributed computing paradigms (Big Data frameworks), and community-driven

standardization of pipelines.

• Containerization and Workflow Portability: Container technologies have become nearly ubiquitous as a solution for reproducibility and portability. Many papers highlighted the positive impact of containers in simplifying deployment on HPC. By using containers (especially with Singularity on HPC), researchers package all software dependencies with the pipeline, eliminating the “it works on one cluster but not another” problem [3]. Containers also make it easier to run the same pipeline on a cloud or a local machine. An emerging best practice is for pipeline authors to provide pre-built container images (often via Docker Hub or Quay) or Conda environment files alongside their workflow. For example, the nf-core repository ensures every pipeline has an accompanying Docker/Singularity image, which users on HPC can invoke to get a consistent environment. Additionally, workflow languages have evolved to integrate containers as a first-class concept: both CWL and WDL include syntax to specify container images for each task, and Nextflow’s configuration can pin specific containers per process.

HPC-Aware Workflow Engines and Scheduling: Recognizing the limitations of traditional HPC scheduling for thousands of tiny tasks, both workflow developers and HPC centers are devising solutions. One approach is the integration of job grouping or pilot jobs - Nextflow’s DX executor or IBM’s LSF Flow - that bundle many tasks into a single job submission to reduce scheduler strain. Another solution is the development of new workflow-native schedulers. The review included one study of Swift/T, a language that allows implicit parallelism and uses its own runtime to map tasks onto MPI ranks across an HPC allocation [11]. Swift/T, coming from the supercomputing world, demonstrated the ability to handle extreme scales and is being applied to bioinformatics for cases where unprecedented parallelism is needed. However, Swift/T requires more specialized knowledge, so its adoption in bioinformatics is still limited. Meanwhile, mainstream WfMSs are improving their HPC integrations: Nextflow has added features to better handle array jobs and has a resource auto-scaling capability that can also apply to on-prem clusters. Some HPC centers have begun providing workflow as a service interfaces - for example, implementing the GA4GH Workflow Execution Service (WES) API [25]. This allows users to submit a workflow (written in WDL or CWL) to a unified endpoint, and the HPC backend takes care of executing it with appropriate scheduling. Such abstractions, along with emerging meta-schedulers like Kubernetes on HPC, blur the line between traditional batch HPC and more flexible cloud-like execution, making it easier to scale pipelines. Notably, one 2021 study systematically evaluated Nextflow, WDL with Cromwell, CWL, and Swift/T on the same tasks across HPC and cloud [11]. They found that Nextflow and WDL generally offered the best mix of usability and performance on HPC, with Nextflow being «one of the most mature» solutions and providing desirable features like portability and provenance out-of-the-box. WDL was praised for its intuitive syntax for bioinformaticians and strong community support in genomics, although it required more effort to set up the engine in HPC mode. CWL’s strength was in standardization and interoperability, but it sometimes suffered from fewer features depending on the chosen engine. These comparisons are driving further improvements; for example, Nextflow’s recent DSL2 update added modularization and WDL is being optimized for better scheduler interaction.

Big Data Frameworks and HPC-Cloud Hybrid Solutions: An interesting thread in the literature is the exploration of big data processing frameworks, like Hadoop MapReduce, Apache Spark, and Dask, for bioinformatics pipelines. Traditional HPC excels at running many independent tasks or large tightly-coupled MPI jobs, but some bioinformatics tasks can be reframed as big data problems. A few studies reported on using Spark for genomic analyses - for example, mapping short reads and variant calling on a Spark cluster versus an HPC cluster [27]. In one case, a Hadoop/Spark implementation of a sequence alignment and variant-calling pipeline achieved better scalability and comparable runtime to an equivalent HPC pipeline, largely due to more efficient data distribution [28]. However, these approaches require rewriting parts of pipelines for the new paradigm, often using Java and Scala for Spark, which is a barrier. A compromise emerging solution is hybrid HPC-cloud workflows. Some pipelines now leverage cloud resources dynamically when HPC queues are long or certain tools run better on cloud instances. For example, a genomics analysis might offload a machine-learning-based variant prioritization step to a cloud GPU service, while running core

alignment on the HPC cluster. Workflow systems are starting to support this: Nextflow's tower and CloudBurst features let a pipeline burst to cloud on demand. Federated computing models and container orchestration on HPC are also enabling more dynamic scaling. While not widespread yet, a couple of reviewed articles envision future pipelines that seamlessly use a mix of on-premise HPC, public cloud, and even edge computing, orchestrated by workflow tools that choose the optimal execution venue for each task. This could mitigate some HPC limitations and provide virtually unlimited scalability for embarrassingly parallel tasks.

Community Standardization (Shared Workflows and Best Practices): A very significant development in recent years, highlighted in multiple sources, is the rise of community-curated workflow collections. nf-core is a notable example - a community effort that curates high-quality Nextflow pipelines for various common bioinformatics analyses [26]. These pipelines are developed collaboratively, reviewed, and adhere to consistent standards. For HPC users, nf-core pipelines provide ready-made solutions that have been tested on different systems, reducing the effort needed to set up a new pipeline. Many of the included studies, instead of writing a new pipeline from scratch, referenced using or adapting an nf-core pipeline - for example, using nf-core and mag for metagenomic assembly or nf-core/rnaseq for transcriptomics. This trend greatly improves reproducibility and reduces duplicated effort. Similarly, the Galaxy Tool Shed and WorkflowHub serve as repositories where scientists can find and reuse workflows [22]. The GA4GH Tool Registry Service (TRS) standard underpins some of these platforms, enabling tools and workflows to be shared in a standardized way [25]. Overall, the community is moving toward "write once, run anywhere" pipelines. By sharing best-practice workflows, researchers can apply the same pipeline on their local supercomputer or on a cloud platform, confident that results will be comparable. This also facilitates benchmarking, like one study used the nf-core variant calling pipeline to systematically compare performance on an HPC cluster versus an AWS cloud instance, demonstrating comparable accuracy but noting the cloud incurred higher cost for similar runtime. The knowledge gained from such comparisons feeds back into best practices.

Specialized Pipeline Implementations: The maturation of HPC bioinformatics is evident in domain-specific automated platforms that demonstrate best practices at scale. VPipe exemplifies this trend as an automated platform that has processed over 17,500 clinical specimens for viral pathogen characterization, showcasing how specialized pipelines can achieve production-level throughput [29]. At the consortium level, the ENCODE project's uniform analysis pipelines represent the gold standard for standardized computational methodologies, processing over 19,000 functional genomics experiments using Docker and WDL to ensure reproducible results across diverse HPC and cloud environments [30].

• Provenance and Data Standards: Alongside workflows, there's recognition of the need for standardizing the reporting of analyses and outputs to improve interoperability. Some emerging solutions include capturing the RO-Crate metadata for workflows and using common output formats so that results from different pipelines can be aggregated. While not the central focus of most included papers, a few touched on these aspects, indicating that as pipelines become more automated, they are also becoming more transparent. We see early steps toward this with consortia adopting uniform pipelines, in the paper by Ahmed et al. [11] was essentially about what features matter when choosing a WfMS for production genomics in a clinical setting. They concluded that factors such as language expressiveness, modularity, and community adoption should guide tool choice, and noted that as the community's needs evolve, WfMSs must also evolve - especially those with open, permissive licenses that allow widespread contribution and commercial support.

Table 2. Examples of automated bioinformatics pipelines in different domains, their implementation, and HPC considerations.

Domain & Pipeline	**Workflow Framework**	**HPC Implementation**	**Reproducibility & Notes**
Disease Genomics - Variant Calling (e.g. GATK Best	WDL (Cromwell engine) or Nextflow (nf-	Runs on clusters (multi-node) with scatter-gather parallelism (per-	Broad Institute provides Docker images for GATK; pipeline versions are fixed for consistency. Proven in

Practices pipeline)	core/variantcalling)	chromosome or per-sample processing). Requires careful job concurrency control to avoid scheduler overload []. HPC nodes with high memory used for genome processing.	both research and clinical diagnostic settings for identifying mutations.
Cancer Transcriptomics - RNA-seq Expression (e.g. nf-core/rnaseq)	Nextflow (DSL2) via nf-core pipeline	Typically run on HPC clusters using ~16-32 threads per sample for alignment (STAR/Hisat2) and quantification, across dozens of samples in parallel. Utilizes Slurm executor for job submission.	nf-core/rnaseq uses containerized tools and outputs standardized QC reports. Ensures identical processing across labs. HPC accelerates turnaround for time-sensitive projects (e.g. tumor RNA profiling for precision medicine).
Proteomics - Mass Spectrometry Pipeline (e.g. MoTrPAC prot pipeline)	WDL or Snakemake (with MS tools integrated)	HPC or cloud HPC used to search large protein databases with parallel searches. Tasks like peptide-spectrum matching distributed across nodes. Potential use of GPU nodes for deep learning-based spectra analysis.	Use of containers or Conda for tools like SearchGUI/PeptideShaker. Emphasis on reproducibility since analyses may be regulatory-grade. One project adopted WDL to describe the proteomics workflow, enabling uniform execution on different HPC sites.
Metagenomics - Metagenome Assembly & Binning (e.g. nf-core/mag)	Nextflow (nf-core pipeline)	Runs hybrid assemblies (short + long reads) on HPC, using many CPUs and high memory per job (megahit/SPAdes, etc.). Utilizes Nextflow's ability to resume and parallelize by sample or contig. Can scale out to cloud for especially large data sets [].	nf-core/mag pipeline comes with Docker/Singularity images for all tools. Reproducibility facilitated by fixed software versions. Has been used in large microbiome projects, demonstrating portability across HPC clusters and cloud platforms.
Single-cell Genomics - scRNA-seq or ATAC-seq pipeline	Snakemake or Nextflow (community pipelines like Cell Ranger wrappers)	HPC used due to heavy memory and CPU needs for processing hundreds of millions of reads. Parallelizes by sample and by task (alignment, barcode processing, etc.). Some pipelines use array jobs for each sample.	Often relies on 10x Genomics provided pipelines (which are internally parallelized). Researchers wrap these in Snakemake to handle multiple samples. Environments managed with Conda for reproducibility. HPC ensures that large single-cell datasets (millions of cells) can be processed in reasonable time.

Across these diverse applications, common themes of HPC optimization emerge - such as splitting workloads into parallel chunks, tuning resource usage per step, and utilizing HPC-specific features like job arrays, high-memory nodes to improve efficiency. The emerging best practices from the literature include: using containers for all analyses, applying workflow frameworks that match the project's needs (Nextflow for complex branching pipelines, Snakemake for simpler linear pipelines or when Python integration is needed), and relying on community-validated pipelines when possible instead of reinventing the wheel. Furthermore, authors stress testing pipelines with synthetic or smaller data first before scaling out to a full cluster run, to avoid costly failures mid-run on large systems.

The literature we surveyed shows that workflow frameworks and containerization practices are now production-grade and widely adopted across domains-from clinical genomics to large-scale metagenomics-rather than experimental curiosities. At the same time, the reviewed works illustrate that the move to large-scale HPC brings persistent technical tensions: scheduler saturation, I/O contention on parallel file systems, heterogeneity of hardware, and the operational complexity of preserving reproducibility across diverse centers.

Several clear trends emerge from the citations included in this review. First, community-driven, standardized pipelines and registries are lowering the barrier to reliable, portable analyses: community collections and best-practice pipelines provide tested templates that can be adapted to local HPC environments, reducing duplicated effort and improving cross-site comparability. Second, containerization and image-based deployment remain a dominant strategy for ensuring environment consistency and provenance on HPC; Singularity/Apptainer and pre-built container images are repeatedly reported as practical solutions for running complex toolchains securely on shared supercomputers. Third, the practical integration of GPU-accelerated or ML-based steps into bioinformatics pipelines has started to produce demonstrable performance gains-especially for tasks like accelerated variant calling or deep-learning–based analyses-but this creates new scheduling and resource-allocation considerations for mixed CPU/GPU workloads.

At the infrastructure and workflow-engine level, several works document progress toward reducing scheduler overhead and making workflows «HPC-aware». Approaches include task bundling or pilot-job strategies, workflow-native runtimes that map tasks onto allocations more efficiently, and improvements in mainstream WfMS executors that better support job arrays and resource packing on Slurm or LSF systems. These developments address one of the central pain points documented in our review: naive submission of very large numbers of small tasks can overwhelm scheduler systems, whereas grouped or hierarchical dispatch reduces load and improves throughput. Relatedly, the adoption of GA4GH-style execution APIs and workflow registries is helping to decouple workflow description from execution backend, improving portability across HPC centers and cloud environments.

I/O and storage remain among the most frequently cited bottlenecks for data-intensive pipelines. Empirical studies of HPC file-system behavior, and reports of using node-local scratch and burst buffers, show that careful data placement and intermediate-file management are necessary to avoid severe slowdowns when thousands of tasks perform concurrent reads/writes. Some groups have explored re-framing certain problems with data-local paradigms-bringing compute to the data with Big Data frameworks (Spark/Hadoop) or using distributed object stores-to mitigate I/O hotspots, with promising results in specific use cases such as parallel short-read mapping or large-scale pileup operations. However, these approaches often require non-trivial code rework and expertise in different programming models, so hybrid strategies are being proposed as pragmatic compromises.

Heterogeneous hardware and fine-grained resource scheduling remain difficult to automate end-to-end. Pipelines frequently contain a mixture of memory-heavy, CPU-bound, and GPU-accelerated stages; specifying per-task resource requirements in workflow descriptions helps, but efficient packing and dynamic placement across heterogeneous node types still depend on sophisticated scheduler features or custom runtimes. New tooling such as SLUR(M)-py and related resource-prediction approaches attempt to close this gap by estimating runtime and resource needs to guide packing and reduce waste on multi-architecture systems. These methods show promise but are not yet ubiquitous.

Reproducibility and provenance-already central themes in pipeline literature-continue to improve through combined use of containers, workflow metadata capture, and community standards for workflow sharing. Large consortium pipelines (for example, ENCODE-style uniform analyses and other consortium implementations) demonstrate how standardized, containerized pipelines can produce highly reproducible outputs across HPC and cloud deployments, and how archiving of images, workflow definitions, and logs supports auditability for clinical or regulatory contexts. Still, the literature also highlights practical impediments such as container-pull storming at scale, registry access restrictions on some HPC centers, and the need for light-weight image variants tailored for HPC to reduce network strain.

The intersection of bioinformatics workflows with machine learning is a rapidly growing area. Several cited works present ML-augmented pipelines or GPU-accelerated implementations that significantly reduce runtimes for compute-intensive steps, but they also show that integrating ML requires explicit support in both the workflow description and the execution environment. As ML becomes a more common pipeline component, workflow systems will need richer primitives to

express and manage heterogeneous resource classes seamlessly.

Usability and training are recurring social challenges: while workflow frameworks abstract many complexities, non-expert users still need curated pipelines, sensible defaults, and accessible interfaces to avoid misuse or inefficient runs. Platforms that provide higher-level orchestration-centralized workflow submission endpoints, GUI layers, or "workflow-as-a-service" front-ends backed by HPC-are emerging as practical mitigations and are documented in multiple studies. Community curation and training remain critical levers for wide adoption and correct usage.

Looking forward, several avenues for future work are well supported by the literature we reviewed. First, workflow engines must scale beyond current scheduler and runtime assumptions to support extreme-scale task counts and exabyte-scale data; hierarchical and adaptive dispatch, tighter integration with next-generation schedulers, and modular decomposition of workflows are probable directions. Second, streaming and long-running workflows-needed for near-real-time clinical genomics and continuous sequencing-will require persistent, fault-tolerant runtime semantics that some experimental runtimes and cloud-native approaches are beginning to explore. Third, provenance and packaging standards (RO-Crate, well-annotated workflow registries) should be further adopted so that workflow executions themselves become first-class, shareable research artifacts. Finally, continued community efforts to produce validated, production-grade pipelines will be essential for reproducible science at scale.

All work here indicate that many historical pain points of running bioinformatics pipelines on supercomputers are being systematically addressed through a combination of containerization, improved workflow engines and runtimes, hybrid HPC–big-data paradigms, and community standardization. Nonetheless, operational realities-file-system behavior at scale, scheduler policy constraints, and hardware heterogeneity-persist as active areas for engineering innovation and cross-disciplinary collaboration. As exascale systems and more heterogeneous compute become widely available, continued co-evolution of HPC architectures and workflow systems will be necessary to fully realize the promise of automated bioinformatics at planetary scale.

In reflecting on the limitations of this review, one must note that the field is rapidly evolving. New versions of workflow tools and new pipelines are emerging continuously. Our review captures a snapshot up to late 2025, but the trends suggest continued growth and innovation. We also relied on published literature. There is a wealth of practical knowledge in the form of blog posts, documentation, and forum discussions that are not captured in formal publications. These informal sources often contain tips to overcome HPC challenges. However, we focused on peer-reviewed or well-documented sources to ensure reliability. Another limitation is that our synthesis is qualitative, more measuring the «best» pipeline framework or quantifying performance advantages is highly context-dependent. Some cited benchmarks provided direct comparisons, but the choice of WfMS often depends on user preference and specific project needs as much as on raw performance.

Conclusion

Automated bioinformatics pipelines have become indispensable for extracting knowledge from the deluge of data in genomics, transcriptomics, proteomics, and beyond. This review shows that workflow systems deployed on supercomputers are rising to meet the challenge of large-scale data analysis, with an ecosystem of tools enabling complex analyses to run reproducibly and efficiently. Nextflow, Snakemake, WDL/CWL, and similar frameworks provide the backbone for modern pipelines, allowing researchers to leverage HPC clusters and cloud HPC with relative ease compared to a decade ago. These pipelines power critical scientific and medical advances - from pinpointing genetic variants in rare diseases to profiling microbial communities in global health studies - by delivering results on timescales that only massive parallel computation can achieve.

At the same time, running pipelines on HPC is not simply a matter of throwing software at big iron. Key challenges such as scheduler saturation, I/O bottlenecks, and environment consistency must be addressed to fully capitalize on HPC capabilities. The community has responded with innovative solutions: containerization to encapsulate software and ensure reproducibility, improved workflow engines and scheduling techniques to handle massive task arrays, and even re-imagining pipelines with big data frameworks for certain use cases. Collaborative efforts like nf-core exemplify how the

field is coalescing around shared best practices, improving both reliability and performance of workflows across different HPC platforms.

In conclusion, the marriage of bioinformatics pipelines with supercomputing is maturing. The path forward will likely involve even tighter integration between workflow systems and HPC resource managers, greater use of hybrid cloud-HPC strategies, and ongoing standardization of both methods and metadata. For researchers, investing time in learning and adopting these automated workflows yields rich dividends: faster turnaround, higher reproducibility, and the ability to tackle questions that were once intractable due to data scale. For HPC providers, supporting diverse workflow tools and developing infrastructure that accommodates the unique needs of bioinformatics will be crucial. The emerging solutions documented here suggest that many of the historical pain points are being resolved one by one. As we enter the exascale era, automated bioinformatics pipelines - empowered by HPC - are poised to accelerate scientific discovery, transforming raw data into insights at unprecedented speed and scale.

Acknowledgement. This research has been supported by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. BR28713313).

References

1. Stephens ZD, Lee SY, Faghri F, Campbell RH, Zhai C, Efron MJ, et al. (2015) Big Data: Astronomical or Genomical? PLoS Biol 13(7): e1002195. https://doi.org/10.1371/journal.pbio.1002195

2. Zhou, Y., Kathiresan, N., Yu, Z., Rivera, L. F., Thimma, M., Manickam, K., Chebotarov, D., Mauleon, R., Chougule, K., Wei, S., Gao, T., Green, C. D., Zuccolo, A., Ware, D., Zhang, J., … & Wing, R. A. (2024). A high-performance computational workflow to accelerate GATK SNP detection across a 25-genome dataset. BMC Biology, 22, Article 13. https://doi.org/10.1186/s12915-024-01820-5

3. Djaffardjy, M., Marchment, G., Sebé, C., Blanchet, R., Belhajjame, K., Gaignard, A., Lemoine, F., & Cohen-Boulakia, S. (2023). Developing and reusing bioinformatics data analysis pipelines using scientific workflow systems. Computational and Structural Biotechnology Journal, 21, 2075–2085. https://doi.org/10.1016/j.csbj.2023.03.003

4. Evangelidis, T., & van der Velde, J. (2025). Empowering bioinformatics communities with Nextflow and nf-core. Genome Biology, 26, Article 228. https://doi.org/10.1186/s13059-025-03673-9

5. Amstutz, P. (Ed.), Crusoe, M. R. (Ed.), Tijanić, N. (Ed.), Chapman, B., Chilton, J., Heuer, M., Kartashov, A., Leehr, D., Ménager, H., Nedeljkovich, M., Scales, M., Soiland-Reyes, S., & Stojanovic, L. (2016). Common Workflow Language, v1.0. figshare. https://doi.org/10.6084/m9.figshare.3115156.v2

6. Vivian, J., Rao, A. A., Nothaft, F. A., Ketchum, C., Armstrong, J., Novak, A., … & Paten, B. (2017). Toil enables reproducible, open source, big biomedical data analyses. Nature Biotechnology, 35(4), 314–316. https://doi.org/10.1038/nbt.3772

7. Langer, B. E., Amaral, A., Baudement, M. O., et al. (2025). Empowering bioinformatics communities with Nextflow and nf-core. Genome Biology, 26, Article 228. https://doi.org/10.1186/s13059-025-03673-9

8. Crusoe, M. R., Abeln, S., Iosup, A., Amstutz, P., Chilton, J., Tijanić, N., … & Goble, C. (2021). Methods included: Standardizing computational reuse and portability with the Common Workflow Language. arXiv. https://doi.org/10.48550/arXiv.2105.07028

9. Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., … & Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. BMJ, 372, n71. https://doi.org/10.1136/bmj.n71

10. Ahmed, A. E., Heldenbrand, J., Asmann, Y., Fadlelmola, F. M., Katz, D. S., Kendig, K., … & Zermeno, J. (2019). Genomic workflow management with Swift/T. PLOS ONE, 14(7), e0211608. https://doi.org/10.1371/journal.pone.0211608

11. Ahmed, A. E., Allen, J. M., Bhat, T., Burra, P., Fliege, C. E., Hart, S. N., Heldenbrand, J.

R., Hudson, M. E., Istanto, D. D., Kalmbach, M. T., Kapraun, G. D., Kendig, K. I., Kendzior, M. C., Klee, E. W., Mattson, N., Ross, C. A., Sharif, S. M., Venkatakrishnan, R., Fadlelmola, F. M., & Mainzer, L. S. (2021). Design considerations for workflow management systems use in production genomics research and the clinic. Scientific reports, 11(1), 21680. https://doi.org/10.1038/s41598-021-99288-8

12. Koboldt, D. C., Steinberg, K. M., Larson, D. E., Wilson, R. K., & Mardis, E. R. (2020). Best practices for variant calling in clinical sequencing. Genome Medicine, 12(91). https://doi.org/10.1186/s13073-020-00791-w

13. Larsonneur, E., Mercier, J., Wiart, N., Le Floch, E., Delhomme, O., & Meyer, V. (2018). Evaluating Workflow Management Systems: A Bioinformatics Use Case.

14. Angelova, N., Danis, T., Lagnel, J., Tsigenopoulos, C. S., & Manousaki, T. (2022). SnakeCube: Containerized and automated pipeline for de novo genome assembly in HPC environments. BMC Research Notes, 15, 98. https://doi.org/10.1186/s13104-022-05978-5 SpringerLink+1

15. Genome variant calling workflow implementation and deployment in HPC infrastructure. (2021). In 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE. https://doi.org/10.1109/BIBM52615.2021.9669519 ResearchGate+1

16. Ramos Carneiro, A., Bez, J. L., Osthoff, C., Schnorr, L. M., & Navaux, P. O. A. (2023). Uncovering I/O demands on HPC platforms: Peeking under the hood of Santos Dumont. Journal of Parallel and Distributed Computing, 181, 104768.

17. Visconti, A., Martin, T. C., & Falchi, M. (2018). YAMP: a containerised workflow enabling reproducibility in metagenomics research. GigaScience.

18. Mousavi-Derazmahalleh, M., Stott, A., Lines, R., Peverley, G., Nester, G., Simpson, T., Zawierta, M., De La Pierre, M., Bunce, M., & Christophersen, C. T. (2021). eDNAFlow, an automated, reproducible and scalable workflow for analysis of environmental DNA sequences exploiting Nextflow and Singularity. Molecular ecology resources, 21(5), 1697–1704. https://doi.org/10.1111/1755-0998.13356

19. Budiš, J., Krampl, W., Kucharík, M., Hekel, R., Goga, A., Sitarčík, J., ... & Szemes, T. (2024). SnakeLines: integrated set of computational pipelines for sequencing reads. Journal of Integrative Bioinformatics, 20(3), 20220059.

20. Czech, L., & Exposito-Alonso, M. (2022). grenepipe: a flexible, scalable and reproducible pipeline to automate variant calling from sequence reads. Bioinformatics (Oxford, England), 38(20), 4809–4811. https://doi.org/10.1093/bioinformatics/btac600

21. Wratten, L., Wilm, A., & Göke, J. (2021). Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. Nature Methods, 18, 1161–1168.

22. Jalili, V., Afgan, E., Gu, Q., Clements, D., Blankenberg, D., Goecks, J., Taylor, J., & Nekrutenko, A. (2020). The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2020 update. Nucleic acids research, 48(W1), W395–W402. https://doi.org/10.1093/nar/gkaa434

23. Zhou, J., Zhang, B., Li, G., Chen, X., Li, H., Xu, X., Chen, S., He, W., Xu, C., Liu, L., & Gao, X. (2024). An AI agent for fully automated multi-omic analyses. Advanced Science, 11(44), e2407094. https://doi.org/10.1002/advs.202407094

24. Lang, O., & colleagues. (2022). ScriptManager: an interactive platform for reducing barriers to genomics analysis for novice bioinformaticians. In Proceedings of the PEARC '22: Practice and Experience in Advanced Research Computing (Article No. 3535161). ACM. https://doi.org/10.1145/3491418.3535161

25. Kanitz, A., McLoughlin, M. H., Beckman, L., GA4GH Cloud Workstream, Malladi, V. S., & Ellrott, K. (2024). The GA4GH Task Execution Application Programming Interface: Enabling Easy Multicloud Task Execution. Computing in science & engineering, 26(3), 30–39. https://doi.org/10.1109/mcse.2024.3414994

26. Ewels, P.A., Peltzer, A., Fillinger, S. et al. The nf-core framework for community-curated bioinformatics pipelines. Nat Biotechnol 38, 276–278 (2020). https://doi.org/10.1038/s41587-020-

0439-x

27. Guo, R., Zhao, Y., Zou, Q., Fang, X., & Peng, S. (2018). Bioinformatics applications on Apache Spark. GigaScience, 7(8), giy098. https://doi.org/10.1093/gigascience/giy098

28. Decap, D., de Schaetzen van Brienen, L., Larmuseau, M., Costanza, P., Herzeel, C., Wuyts, R., Marchal, K., & Fostier, J. (2022). Halvade Somatic: Somatic variant calling with Apache Spark. GigaScience, 11, giab094. https://doi.org/10.1093/gigascience/giab094

29. Wagner, D. D., Garry, D., Krueger, S., Cole, S., Nadon, C., & Greig, A. (2022). VPipe: An automated bioinformatics platform for assembly and management of viral next-generation sequencing data. Microbiology Spectrum, 10(2), e02564-21. https://doi.org/10.1128/spectrum.02564-21

30. Hitz, B. C., Jin-Wook, L., Jolanki, O., Kagda, M. S., Graham, K., Sud, P., Gabdank, I., Strattan, J. S., Sloan, C. A., Dreszer, T., Rowe, L. D., Podduturi, N. R., Malladi, V. S., Chan, E. T., Davidson, J. M., Ho, M., Miyasato, S., Simison, M., Tanaka, F., Luo, Y., … Cherry, J. M. (2023). The ENCODE Uniform Analysis Pipelines. bioRxiv : the preprint server for biology, 2023.04.04.535623. https://doi.org/10.1101/2023.04.04.535623

Әдебиеттер тізімі

1. Stephens ZD, Lee SY, Faghri F, Campbell RH, Zhai C, Efron MJ, et al. (2015) Big Data: Astronomical or Genomical? PLoS Biol 13(7): e1002195. https://doi.org/10.1371/journal.pbio.1002195

2. Zhou, Y., Kathiresan, N., Yu, Z., Rivera, L. F., Thimma, M., Manickam, K., Chebotarov, D., Mauleon, R., Chougule, K., Wei, S., Gao, T., Green, C. D., Zuccolo, A., Ware, D., Zhang, J., … & Wing, R. A. (2024). A high-performance computational workflow to accelerate GATK SNP detection across a 25-genome dataset. BMC Biology, 22, Article 13. https://doi.org/10.1186/s12915-024-01820-5

3. Djaffardjy, M., Marchment, G., Sebé, C., Blanchet, R., Belhajjame, K., Gaignard, A., Lemoine, F., & Cohen-Boulakia, S. (2023). Developing and reusing bioinformatics data analysis pipelines using scientific workflow systems. Computational and Structural Biotechnology Journal, 21, 2075–2085. https://doi.org/10.1016/j.csbj.2023.03.003

4. Evangelidis, T., & van der Velde, J. (2025). Empowering bioinformatics communities with Nextflow and nf-core. Genome Biology, 26, Article 228. https://doi.org/10.1186/s13059-025-03673-9

5. Amstutz, P. (Ed.), Crusoe, M. R. (Ed.), Tijanić, N. (Ed.), Chapman, B., Chilton, J., Heuer, M., Kartashov, A., Leehr, D., Ménager, H., Nedeljkovich, M., Scales, M., Soiland-Reyes, S., & Stojanovic, L. (2016). Common Workflow Language, v1.0. figshare. https://doi.org/10.6084/m9.figshare.3115156.v2

6. Vivian, J., Rao, A. A., Nothaft, F. A., Ketchum, C., Armstrong, J., Novak, A., … & Paten, B. (2017). Toil enables reproducible, open source, big biomedical data analyses. Nature Biotechnology, 35(4), 314–316. https://doi.org/10.1038/nbt.3772

7. Langer, B. E., Amaral, A., Baudement, M. O., et al. (2025). Empowering bioinformatics communities with Nextflow and nf-core. Genome Biology, 26, Article 228. https://doi.org/10.1186/s13059-025-03673-9

8. Crusoe, M. R., Abeln, S., Iosup, A., Amstutz, P., Chilton, J., Tijanić, N., … & Goble, C. (2021). Methods included: Standardizing computational reuse and portability with the Common Workflow Language. arXiv. https://doi.org/10.48550/arXiv.2105.07028

9. Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., … & Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. BMJ, 372, n71. https://doi.org/10.1136/bmj.n71

10. Ahmed, A. E., Heldenbrand, J., Asmann, Y., Fadlelmola, F. M., Katz, D. S., Kendig, K., … & Zermeno, J. (2019). Genomic workflow management with Swift/T. PLOS ONE, 14(7), e0211608. https://doi.org/10.1371/journal.pone.0211608

11. Ahmed, A. E., Allen, J. M., Bhat, T., Burra, P., Fliege, C. E., Hart, S. N., Heldenbrand, J.

R., Hudson, M. E., Istanto, D. D., Kalmbach, M. T., Kapraun, G. D., Kendig, K. I., Kendzior, M. C., Klee, E. W., Mattson, N., Ross, C. A., Sharif, S. M., Venkatakrishnan, R., Fadlelmola, F. M., & Mainzer, L. S. (2021). Design considerations for workflow management systems use in production genomics research and the clinic. Scientific reports, 11(1), 21680. https://doi.org/10.1038/s41598-021-99288-8

12. Koboldt, D. C., Steinberg, K. M., Larson, D. E., Wilson, R. K., & Mardis, E. R. (2020). Best practices for variant calling in clinical sequencing. Genome Medicine, 12(91). https://doi.org/10.1186/s13073-020-00791-w

13. Larsonneur, E., Mercier, J., Wiart, N., Le Floch, E., Delhomme, O., & Meyer, V. (2018). Evaluating Workflow Management Systems: A Bioinformatics Use Case.

14. Angelova, N., Danis, T., Lagnel, J., Tsigenopoulos, C. S., & Manousaki, T. (2022). SnakeCube: Containerized and automated pipeline for de novo genome assembly in HPC environments. BMC Research Notes, 15, 98. https://doi.org/10.1186/s13104-022-05978-5 SpringerLink+1

15. Genome variant calling workflow implementation and deployment in HPC infrastructure. (2021). In 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE. https://doi.org/10.1109/BIBM52615.2021.9669519 ResearchGate+1

16. Ramos Carneiro, A., Bez, J. L., Osthoff, C., Schnorr, L. M., & Navaux, P. O. A. (2023). Uncovering I/O demands on HPC platforms: Peeking under the hood of Santos Dumont. Journal of Parallel and Distributed Computing, 181, 104768.

17. Visconti, A., Martin, T. C., & Falchi, M. (2018). YAMP: a containerised workflow enabling reproducibility in metagenomics research. GigaScience.

18. Mousavi-Derazmahalleh, M., Stott, A., Lines, R., Peverley, G., Nester, G., Simpson, T., Zawierta, M., De La Pierre, M., Bunce, M., & Christophersen, C. T. (2021). eDNAFlow, an automated, reproducible and scalable workflow for analysis of environmental DNA sequences exploiting Nextflow and Singularity. Molecular ecology resources, 21(5), 1697–1704. https://doi.org/10.1111/1755-0998.13356

19. Budiš, J., Krampl, W., Kucharík, M., Hekel, R., Goga, A., Sitarčík, J., ... & Szemes, T. (2024). SnakeLines: integrated set of computational pipelines for sequencing reads. Journal of Integrative Bioinformatics, 20(3), 20220059.

20. Czech, L., & Exposito-Alonso, M. (2022). grenepipe: a flexible, scalable and reproducible pipeline to automate variant calling from sequence reads. Bioinformatics (Oxford, England), 38(20), 4809–4811. https://doi.org/10.1093/bioinformatics/btac600

21. Wratten, L., Wilm, A., & Göke, J. (2021). Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. Nature Methods, 18, 1161–1168.

22. Jalili, V., Afgan, E., Gu, Q., Clements, D., Blankenberg, D., Goecks, J., Taylor, J., & Nekrutenko, A. (2020). The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2020 update. Nucleic acids research, 48(W1), W395–W402. https://doi.org/10.1093/nar/gkaa434

23. Zhou, J., Zhang, B., Li, G., Chen, X., Li, H., Xu, X., Chen, S., He, W., Xu, C., Liu, L., & Gao, X. (2024). An AI agent for fully automated multi-omic analyses. Advanced Science, 11(44), e2407094. https://doi.org/10.1002/advs.202407094

24. Lang, O., & colleagues. (2022). ScriptManager: an interactive platform for reducing barriers to genomics analysis for novice bioinformaticians. In Proceedings of the PEARC ’22: Practice and Experience in Advanced Research Computing (Article No. 3535161). ACM. https://doi.org/10.1145/3491418.3535161

25. Kanitz, A., McLoughlin, M. H., Beckman, L., GA4GH Cloud Workstream, Malladi, V. S., & Ellrott, K. (2024). The GA4GH Task Execution Application Programming Interface: Enabling Easy Multicloud Task Execution. Computing in science & engineering, 26(3), 30–39. https://doi.org/10.1109/mcse.2024.3414994

26. Ewels, P.A., Peltzer, A., Fillinger, S. et al. The nf-core framework for community-curated bioinformatics pipelines. Nat Biotechnol 38, 276–278 (2020). https://doi.org/10.1038/s41587-020-

0439-x

27. Guo, R., Zhao, Y., Zou, Q., Fang, X., & Peng, S. (2018). Bioinformatics applications on Apache Spark. GigaScience, 7(8), giy098. https://doi.org/10.1093/gigascience/giy098

28. Decap, D., de Schaetzen van Brienen, L., Larmuseau, M., Costanza, P., Herzeel, C., Wuyts, R., Marchal, K., & Fostier, J. (2022). Halvade Somatic: Somatic variant calling with Apache Spark. GigaScience, 11, giab094. https://doi.org/10.1093/gigascience/giab094

29. Wagner, D. D., Garry, D., Krueger, S., Cole, S., Nadon, C., & Greig, A. (2022). VPipe: An automated bioinformatics platform for assembly and management of viral next-generation sequencing data. Microbiology Spectrum, 10(2), e02564-21. https://doi.org/10.1128/spectrum.02564-21

30. Hitz, B. C., Jin-Wook, L., Jolanki, O., Kagda, M. S., Graham, K., Sud, P., Gabdank, I., Strattan, J. S., Sloan, C. A., Dreszer, T., Rowe, L. D., Podduturi, N. R., Malladi, V. S., Chan, E. T., Davidson, J. M., Ho, M., Miyasato, S., Simison, M., Tanaka, F., Luo, Y., … Cherry, J. M. (2023). The ENCODE Uniform Analysis Pipelines. bioRxiv: the preprint server for biology, 2023.04.04.535623. https://doi.org/10.1101/2023.04.04.535623

СУПЕРКОМПЬЮТЕРЛЕРДЕГІ АВТОМАТТАНДЫРЫЛҒАН БИОИНФОРМАТИКА ҚҰБЫРЛАРЫ: ҚИЫНДЫҚТАР ЖӘНЕ ЖАҢА ШЕШІМДЕР

АШИМГАЛИЕВ М.[1], МУСАБЕК М.[2], МАТКАРИМОВ Б.[1], ЖУМАДИЛЛАЕВА А.К.[1*]

Ашимгалиев Медет[1] – PhD, оқытушы, компьютерлік және бағдарламалық қамтамасыз ету кафедрасы, ақпараттық технологиялар факультеті, Л.Н.Гумилев атындағы Еуразия ұлттық университеті, Астана, Қазақстан.
E-mail: ashimgaliyev.medet@gmail.com, https://orcid.org/0009-0003-9829-6187
Мусабек Мирас[2] – Аға оқытушы, жасанды интеллект және деректер туралы ғылымдар мектебі, Астана ИТ университеті, Астана қ., Қазақстан
E-mail: miras.k@astanait.edu.kz, https://orcid.org/0009-0009-2353-3524
Маткаримов Бақыт[1] – Техника ғылымдарының докторы, профессор, оқытушы және ғылыми қызметкер, жасанды интеллект технологиясы кафедрасы, ақпараттық технологиялар факультеті, Л.Н.Гумилев атындағы Еуразия ұлттық университеті, Астана қ., Қазақстан
E-mail: bakhyt.matkarimov@gmail.com, https://orcid.org/0000-0003-0775-7324
***Жумадиллаева Айнур Канадиловна[1]** - Техника ғылымдарының кандидаты, қауымдастырылған профессор, компьютерлік және бағдарламалық қамтамасыз ету кафедрасы, ақпараттық технологиялар факультеті, Л.Н.Гумилев атындағы Еуразия ұлттық университеті, Астана қ., Қазақстан.
E-mail: Ainur.Zhumadillayeva@astanait.edu.kz, https://orcid.org/0000-0003-1042-0415

Аңдатпа. Жоғары өнімді биологиялық деректерді генерациялау жоғары өнімді есептеу (hpc) жүйелері мен суперкомпьютерлерде автоматтандырылған биоинформатика құбырларын енгізуге түрткі болды. prisma нұсқауларына сәйкес құрастырылған бұл жүйелі шолу 2018 және 2025 жылдар аралығында жарияланған 101 зерттеуді қорытындылайды, онда геномика, транскриптомика, протеомика және метагеномика салаларында hpc орталарында орналастырылған жұмыс процесін басқару жүйелері (wfms) зерттеледі. біз nextflow, snakemake, wdl және cwl сияқты қалыптасқан платформаларды талдап, енгізудегі қиындықтар мен жаңа шешімдерді құжаттадық. негізгі қиындықтарға жаппай параллелизмге байланысты жоспарлағыштың шамадан тыс жүктелуі, ортақ файлдық жүйелердегі енгізу/шығару кедергілері, ресурстардың гетерогенді таралуы және әртүрлі есептеу орталарындағы қайталанымдылық жатады. docker және singularity арқылы контейнерлеу портативтілік пен қайталанымдылықты қамтамасыз етудің басым шешімі ретінде пайда болды. nf-core сияқты қауымдастық бастамалары ең жақсы тәжірибелерге негізделген кураторлық құбырларды ұсыну арқылы осы технологияларды енгізуді жеделдетті. заманауи шешімдерге hpc-қа бейімделген жоспарлау стратегиялары, гибридті бұлт және hpc архитектуралары, сондай-ақ машиналық оқытуға негізделген талдау үшін gpu интеграциясы кіреді. күрделі көп сатылы талдауларды автоматтандырудағы айтарлықтай жетістіктерге қарамастан, жұмыс процесі жүйелері мен hpc инфрақұрылымының үздіксіз бірлескен эволюциясы экзамасштабты деректер көлемдерін өңдеу және масштабта толық қайталанатын есептеу биологиясына қол жеткізу үшін маңызды болып қала береді.

Түйін сөздер: биоинформатика құбырлары, жоғары өнімді есептеулер, жұмыс процесін басқару жүйелері, контейнерлеу, қайталанымдылық, масштабталу.

АВТОМАТИЗИРОВАННЫЕ БИОИНФОРМАТИЧЕСКИЕ КОНВЕЙЕРЫ НА СУПЕРКОМПЬЮТЕРАХ: ПРОБЛЕМЫ И НОВЫЕ РЕШЕНИЯ

АШИМГАЛИЕВ М.[1], МУСАБЕК М.[2], МАТКАРИМОВ Б.[1], ЖУМАДИЛЛАЕВА А.К.[1*]

Ашимгалиев Медет[1] – PhD, преподаватель, кафедра компьютерной и программной инженерии, факультет информационных технологий, Евразийский национальный университет имени Л. Н. Гумилева, г. Астана, Казахстан.
E-mail: ashimgaliyev.medet@gmail.com, https://orcid.org/0009-0003-9829-6187
Мусабек Мирас[2] – Старший преподаватель, школа искусственного интеллекта и обработки данных, Астана ИТ Университет, г. Астана, Казахстан
E-mail: miras.k@astanait.edu.kz, https://orcid.org/0009-0009-2353-3524
Маткаримов Бақыт[1] – Доктор технических наук, профессор, преподаватель и научный сотрудник, кафедра технологий искусственного интеллекта, факультет информационных технологий, Евразийский национальный университет имени Л. Н. Гумилева, г. Астана, Казахстан
E-mail: bakhyt.matkarimov@gmail.com, https://orcid.org/0000-0003-0775-7324
***Жумадиллаева Айнур Канадиловна[1]** - Кандидат технических наук, ассоциированный профессор, кафедра компьютерной и программной инженерии, факультет информационных технологий, Евразийский национальный университет имени Л. Н. Гумилева, г. Астана, Казахстан.
E-mail: Ainur.Zhumadillayeva@astanait.edu.kz, https://orcid.org/0000-0003-1042-0415

Аннотация. Высокопроизводительное генерирование биологических данных стимулировало внедрение автоматизированных биоинформатических конвейеров в высокопроизводительных вычислительных системах (HPC) и суперкомпьютерах. В этом систематическом обзоре, составленном в соответствии с руководящими принципами PRISMA, обобщены 30 исследование, опубликованное в период с 2018 по 2025 год, с целью изучения систем управления рабочими процессами (WfMS), развернутых в средах HPC в областях геномики, транскриптомики, протеомики и метагеномики. Мы проанализировали известные платформы, включая Nextflow, Snakemake, WDL и CWL, задокументировав проблемы их внедрения и новые решения. К ключевым проблемам относятся перегрузка планировщика из-за массивного параллелизма, узкие места ввода-вывода в общих файловых системах, гетерогенное распределение ресурсов и воспроизводимость в различных вычислительных средах. Контейнеризация с помощью Docker и Singularity стала доминирующим решением для обеспечения переносимости и воспроизводимости. Инициативы сообщества, такие как nf-core, ускорили внедрение этих технологий, предоставив тщательно отобранные конвейеры, основанные на передовом опыте. Передовые решения включают стратегии планирования с учетом HPC, гибридные архитектуры облачных вычислений и HPC, а также интеграцию GPU для анализа с использованием машинного обучения. Несмотря на значительный прогресс в автоматизации сложных многоэтапных анализов, продолжающаяся совместная эволюция систем рабочих процессов и инфраструктуры HPC остается необходимой для обработки экзафлопных объемов данных и достижения полностью воспроизводимой вычислительной биологии в больших масштабах.

Ключевые слова: биоинформатические конвейеры, высокопроизводительные вычисления, системы управления рабочими процессами, контейнеризация, воспроизводимость, масштабируемость.