Ardigen

Nextflow-Powered Transformation: Migrating Genomics England's Clinical Workflows

Ardigen: Filip Chrzan, Olha Petryk, Dorota Pikul, Mateusz Rogalski, Przemysław Siekaniec, Błażej Szczerba
Genomics England: Edwin Clark, Luke Paul Buttigieg, Ricardo Humberto Ramirez Gonzalez, Neil Goodgame, Mickey Kim, Tim Richardson

ABSTRACT

To enhance its genomic medicine offerings, Genomics England partnered with Ardigen to migrate its in-house clinical workflow management system, Bertha, to a more flexible solution based on Nextflow. Over the course of several months, we successfully completed this project, gaining valuable experience in the process. This poster outlines the key aspects of our migration effort for the benefit of the reader and future projects involving the transition of workflows to Nextflow.

GENOMICS ENGLAND

Genomics England is a global leader in enabling genomic medicine and research, established by the United Kingdom's Department of Health and Social Care, focused on creating a world where everyone benefits from genomic healthcare.

Building on the 100,000 Genomes Project, it supports the NHS's world-first national whole genome sequencing service and runs the growing National Genomic Research Library, alongside delivering numerous major genomics initiatives. By connecting research and clinical care at a national scale, it enables immediate healthcare benefits and advances for the future.

THE LEGACY SYSTEM - BERTHA

Bertha is a custom, clinical-standard certified workflow manager developed at Genomics England. It is a monolithic system designed for high-throughput genetic sequence analysis. Bertha manages the entire analysis process, from sample reception to the identification of candidate genomic variants. It interacts with various internal APIs, databases, and external services to facilitate this process.

The system runs its workflows on high-performance computing (HPC) clusters using the LSF batch scheduler and requires dedicated hardware for certain tasks. At its core, Bertha operates through a Python-based package, Bertha Compute, which handles bioinformatic tasks using tools like BCFTools and Samtools.

GENIE MIGRATION PROJECT

Genomics England is migrating its custom Bertha workflow manager to a new solution, Genie, which is based on Nextflow and Seqera Platform. This transition is driven by the need to meet ambitious targets, including processing 300,000 samples by 2025, and to support newer use cases more efficiently.

Nextflow provides robustness, adaptability, and compatibility with various infrastructures, including public clouds, which are crucial as data volumes grow. The migration will also improve clinical service reliability, leading to increased demand and generating more data for research.

By adopting off-the-shelf products like Nextflow, Genomics England can focus on its core mission of enabling greater public access to genomic medicine.

CONCLUSIONS

Leveraging Nextflow, along with the strategies and techniques described above, allowed us to successfully migrate several clinical workflows from a complex, monolithic, and actively developed system into a novel, more flexible solution. Prototyping potential migration strategies before starting the actual work helped us make informed, fact-based decisions that shaped the entire effort. Utilizing Nextflow's features such as sub-workflows, adhering to nf-core best practices, and using the nf-test framework enabled us to deliver high-quality workflows. Creating Component Wrapper helped encapsulate aspects of the legacy system that needed to be carried over, resulting in a cleaner architecture. Jasmine, along with the layered testing approach, safeguarded us against bugs and divergence between Bertha and Genie.

The resulting pipelines performed admirably, were quickly adopted by the recipient teams, and have already processed the first batches of patient clinical samples without any errors.

CHOOSING A MIGRATION STRATEGY

The first phase of the Genie migration project involved designing the migration process, during which Genomics England proposed three potential migration strategies. To evaluate these options, we were tasked with creating a proof-of-concept (PoC) for each strategy. These PoCs involved quickly prototyping the migration of a single workflow component (equivalent to a Nextflow module) from Bertha to Nextflow. Once completed, we compared the complexity, advantages, challenges, and time needed for each approach. The insights gathered from these PoCs were then used to inform the decision on which migration strategy to adopt. This approach allowed us to evaluate the available options based on facts rather than assumptions, enabling us to choose the solution best suited for Genomics England's business needs.

STRATEGY A - FULL REWRITE

Strategy A involved rewriting all Bertha workflows and functionality from scratch directly in Nextflow. The key advantage was the ability to leverage all of Nextflow's features, such as resource management and parallelization, while eliminating Bertha's technical debt and bringing everything up to industry standards. However, this approach posed significant challenges, including a high risk of divergence between the still-active Bertha system and the newly developed Nextflow workflows.

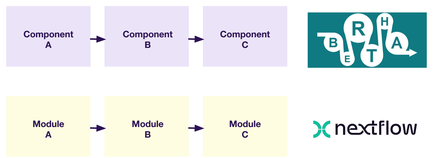

Figure 1: Strategy A completely discards Bertha components.

STRATEGY B - LIFT AND SHIFT

Strategy B, the "lift and shift" approach, involved using the existing Bertha logic without modification by running Bertha components within Nextflow workflows, with each Nextflow module calling Bertha code through a Python wrapper script. The primary benefit of this strategy was its ability to minimize the risk of divergence between Bertha and Nextflow, as Bertha's code was still used without modification. It was also the quickest to implement. However, it did not utilize many of Nextflow's advanced features and retained Bertha's legacy code and associated technical debt.

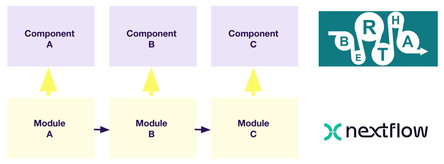

Figure 2: In **Strategy B** Nextflow modules execute Bertha components.

STRATEGY C - EXTRACT & REUSE LOGIC

Strategy C involved extracting the business logic from Bertha into a new Python library and refactoring Bertha to depend on this library. Both Bertha and Nextflow would then use this library, ensuring no divergence between the two systems. This approach offered a clean, testable system architecture, making it easy to port pipelines to other workflow managers in the future and providing an opportunity to remove technical debt. However, the extraction process was risky and potentially time-consuming.

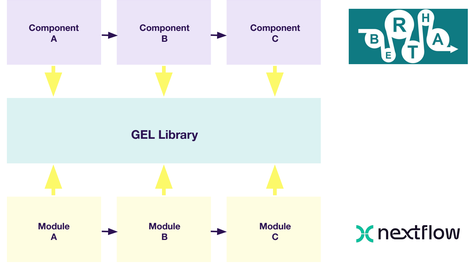

Figure 3: Strategy C: extracting business logic to a shared library.

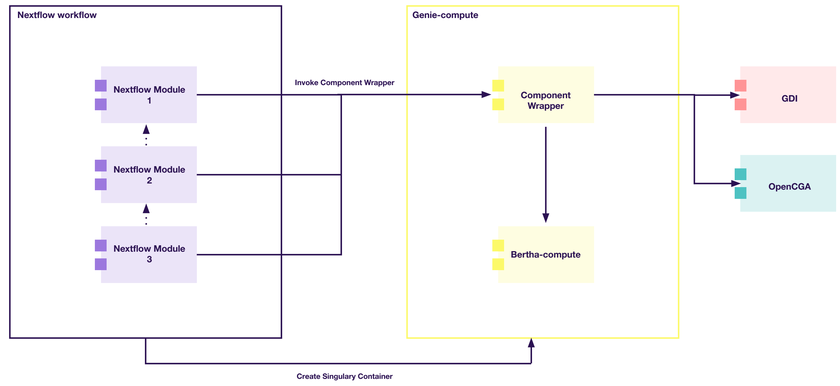

Figure 4: high level overview of Genie. A migrated Nextflow workflow (blue) consists of several modules. When running, each of these modules uses the **genie-compute** Singularity container (yellow) to invoke **Component Wrapper**. Component Wrapper executes the Bertha workflow component corresponding to the Nextflow module and provides access to external services (green).

LIFT & SHIFT APPROACH

After the final evaluation of the migration prototypes, Genomics England decided to proceed with strategy B—the "lift and shift" approach. The primary goal was to implement an initial version of Genie as quickly as possible while minimizing the risk of divergence from Bertha.

In this strategy, the new Genie workflows were constructed from Nextflow modules encapsulating Bertha workflow components and executing them via a custom Python wrapper, called **Component Wrapper**. While this approach reduced the likelihood of Genie deviating from Bertha, some risk still existed. To mitigate this, we developed **Jasmine**, a testing framework designed to compare the outputs of both systems and ensure they produced equivalent results. We also used **containerization** to improve test reproducibility and maintain a controlled development environment.

CONTAINERIZATION

Containerization involves packaging software code along with the necessary operating system libraries and dependencies to create a single lightweight executable, which runs consistently across any infrastructure. We leveraged this approach by creating genie-compute, a Singularity image that contains Bertha components, the Component Wrapper and all required dependencies. This image was used to run all Genie Nextflow workflows. This approach greatly enhanced reproducibility for running our tests and stabilized our development environment. It also gave us complete control over the version of Bertha being used by Genie workflows, which was crucial, as Bertha was still being actively developed and modified during the migration process.

TESTING

In order to ensure no divergence between Bertha and Genie and provide highest code quality, we used a layered testing approach. All Python code in Component Wrapper was covered by unit and integration tests. Nextflow modules and workflows were tested using nf-test. Jasmine enabled implementing high-level functional black-box tests. We also created special input datasets for the workflows, consisting of minimized genetic data, called **tiny-genome** to greatly reduce test execution time. All these tests were automated using GitLab CI. Along with regularly performed manual tests, they formed a robust test suite that helped us maintain the highest quality in our workflows.

COMPONENT WRAPPER

Component Wrapper is a Python program specifically designed to import and run Bertha components within the Genie system. It is distributed in a Singularity image called genie-compute, which is built using a GitLab CI process. Component Wrapper is executed through its entry-point script, run_bertha_component.py, which handles the configuration and parameter generation required to run Bertha components. All Nextflow modules in Genie call this script. Component Wrapper also contains special behaviors for certain components, such as sending HTTP requests directly to external services. By creating Component Wrapper we encapsulated various hidden dependencies and special behaviors of Bertha components into one, testable package, greatly improving the system's architecture.

JASMINE

Jasmine is a custom testing framework built by our team to create an environment capable of running both Bertha components and Genie Nextflow workflows, ensuring their consistency. Originally a collection of bash scripts, Jasmine has evolved into a full Python program. Jasmine operates by first setting up a test environment with Docker containers that run various services required by the workflows, such as OpenCGA, GDI, and Bertha services. Through its CLI, users can run tests that compare the behavior of Bertha and Genie. While Jasmine does not directly verify the correctness of the components themselves, it ensures that Bertha and Genie produce the same outputs by comparing the end states of various services and output files. Test cases are defined within the workflow repository using JSON config files, and test input data is organized into directories. Jasmine also supports grouping multiple components into subsections to streamline testing, but one-component test cases can be run as well. Thanks to Jasmine, we were able to identify various bugs early and ensure the correctness of our migration process.

SUB-WORKFLOWS

To optimize development speed and expand testing scopes, we leveraged Nextflow's sub-workflow feature, which allows workflows to be composed of smaller groups of modules. Using sub-workflows helped reduce complexity by dividing the main workflow into logical components and allowed us to parallelize the migration work by developing multiple sub-workflows simultaneously. Additionally, testing individual sub-workflows enabled us to create smaller, more manageable comparison tests.

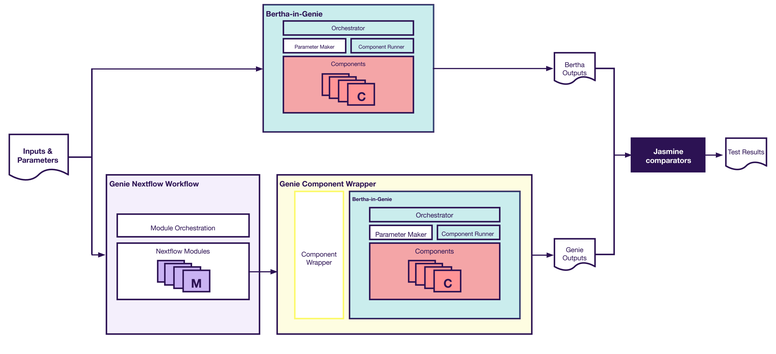

Figure 5: overview of a **Jasmine comparison test**. Identical input data and parameters are fed first to a set of Bertha workflow components (first run) and then to the equivalent migrated Genie workflow (second run). After each run completes, the end state of the system is collected, including workflow outputs and state of external services. Finally, both states are compared. The test passes if no meaningful differences are identified.

NEXTFLOW OVERVIEW

Nextflow is an open-source workflow management system that simplifies the creation and execution of scalable, reproducible data analysis pipelines. It allows users to run complex workflows across different environments, from local machines to cloud platforms, with built-in support for parallelization and containerization. Seqera Platform, built on Nextflow, offers enterprise-level tools for pipeline automation, collaboration, and large-scale workflow optimization, making it easier for teams to manage, scale, and monitor data workflows in real time. Together, they provide a powerful solution for modern data-intensive computing.

ADVANTAGES FOR GENIE MIGRATION

Nextflow is an ideal choice for the Genie migration due to its flexibility, scalability, and cloud compatibility. Its modular design and support for hybrid infrastructures (HPC and cloud) make it highly adaptable. This allows for an incremental migration strategy, with a gradual transition from Bertha's monolithic structure to Genie, which is crucial for minimizing the risk of functional divergence between the two systems—a key concern for a live clinical service. Nextflow will also enable Genie to efficiently manage growing volumes of genomic data, while its seamless integration with bioinformatics tools and modular workflows will provide Genie with greater longevity by simplifying future adaptations. Additionally, its strong community support and alignment with clinical standards will allow Genomics England to focus on delivering genomic medicine solutions while benefiting from ready-made innovations and improvements.

NF-CORE

nf-core is a collaborative, community-driven initiative that provides a collection of high-quality, standardized, and reusable workflows built using Nextflow. These workflows adhere to strict development guidelines curated by the Nextflow and bioinformatics communities, ensuring they are well-documented and of high quality. As a result, nf-core can serve as a perfect source of proven best-practices, working examples and solutions for implementing new workflows. During the migration work we often referenced nf-core resources for inspiration and guidance.

NF-TEST

nf-test is a testing framework designed to validate Nextflow workflows. Developed as part of the nf-core initiative, nf-test allows developers to run unit tests on workflows, checking key components such as configuration parameters, input/output files, and tool dependencies. By integrating with continuous integration (CI) platforms, nf-test helps ensure that pipelines remain functional and up-to-date as they evolve, ultimately enhancing their reliability and development speeds.

NEXTFLOW COMMUNITY

Nextflow has a vibrant community of bioinformaticians, developers, and researchers dedicated to advancing workflow development. This community plays an important role in supporting users through open collaboration, shared knowledge, and best practices. By contributing to documentation, providing troubleshooting help, and offering solutions via forums like Slack, the community fosters an environment where users can learn from each other and improve their workflows. Additionally, the community actively develops initiatives like nf-core, which provide standardized workflows that can be adapted for specific use cases.