NextGen

Genome-Centric Multimodal Data Integration in Personalised Cardiovascular Medicine.

Deliverable 3.8
Accelerated secondary genomic analysis software

Grant Agreement Number: 101136962

Funded by the European Union

The British associated partners of NextGen were funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant agreements No. 10039097, No. 10104323]

Project funded by

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

NextGen	
Project full title	Next Generation Tools for Genome-Centric Multimodal Data Integration In Personalised Cardiovascular Medicine
Call identifier	HORIZON-HLTH-2023-TOOL-05-04
Type of action	RIA
Start date	01/ 01/ 2024
End date	31/12/2027
Grant agreement no	101136962

Funding of associated partners
The Swiss associated partners of the NextGen project were funded by the Swiss State Secretariat for Education, Research and Innovation (SERI). The British associated partners of NextGen were funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant agreements No 10098097, No 10104323]

D3.8 – Accelerated secondary genomic analysis software report	
Author(s)	Eugenio Marinelli, Lorenzo Tattini
Editor	Raja Appuswamy
Participating partners	EURECOM
Version	1.0
Status	Final
Deliverable date	M18
Dissemination Level	PU - Public
Official date	2024-06-30
Actual date	2024-07-04

Disclaimer

The NEXTGEN consortium consists of the following partners:

No	PARTNER ORGANISATION NAME	ABBREVIATION	COUNTRY
1	UNIVERSITAIR MEDISCH CENTRUM UTRECHT	UMCU	NL
2	HIRO MICRODATACENTERS B.V.	HIRO	NL
3	EURECOM GIE	EURE	FR
4	JOHANN WOLFGANG GOETHE-UNIVERSITAET FRANKFURT AM MAIN	GUF	DE
5	KAROLINSKA INSTITUTET	KI	SE
6	HUS-YHTYMA	HUS	FI
7	UNIVERSITY OF VIRGINIA	UVA	US
8	KLINIKUM RECHTS DER ISAR DER TECHNISCHEN UNIVERSITAT MUNCHEN	TUM-Med	DE
9	HL7 INTERNATIONAL FOUNDATION	HL7	BE
10	MYDATA GLOBAL RY	MYDTA	FI
11	DATAPOWER SRL	DPOW	IT
12	SOCIETE EUROPEENNE DE CARDIOLOGIE	ESC	FR
13	WELLSPAN HEALTH	WSPAN	US
14	LIKE HEALTHCARE RESEARCH GMBH	LIKE	DE
15	NEBS SRL	NEBS	BE
16	THE HUMAN COLOSSUS FOUNDATION	HCF	CH
17	SCUOLA UNIVERSITARIA PROFESSIONALE DELLA SVIZZERA ITALIANA	SUPSI	CH
18	DRUG INFORMATION ASSOCIATION	DIA	CH
19	DPO ASSOCIATES SARL	DPOA	CH
20	QUEEN MARY UNIVERSITY OF LONDON	QMUL	UK
21	EARLHAM INSTITUTE	ERLH	UK

Document Revision History

DATE	VERSION	DESCRIPTION	CONTRIBUTIONS
10/06/2025	0.1	Initial demo of the pipeline presented at the GA meeting in Crete	EURECOM
04/07/2025	1.0	Deliverable report describing the demo	EURECOM

Authors

AUTHOR/EDITOR	ORGANISATION
Eugenio Marinelli	EURECOM
Lorenzo Tattini	EURECOM

Reviewers

REVIEWER	ORGANISATION
Raja Appuswamy	EURECOM

List of terms and abbreviations

ABBREVIATION	DESCRIPTION
GATK	Genome Analysis Toolkit
WGS	Whole Genome Sequencing
WES	Whole Exome Sequencing
GAL	Genomics Acceleration Library
BQSR	Base Quality Score Recalibration
VCF	Variant call format
WP	Work Package

Table of contents

1 Summary

D3.8 is a demo deliverable of the SYCL-Genomics Acceleration Library (SYCL-GAL) that is being developed by EURECOM. The goal of the demo is to build a preliminary version of the library to demonstrate feasibility, and enable integration with other components of NextGen in the context of pilots and pathfinder projects. As planned, a preliminary version of SYCL-GAL has been developed by M18, and demoed at the second NextGen General Assembly meeting to all consortium partners. This document provides a high-level outline of the work done in re-implementation of the GATK secondary analysis pipeline and its optimisation for enhanced performance through multi-threaded CPU-based processing.

2 Overview

GATK is a state-of-the-art tool for processing both whole-genome and whole-exome sequencing (WGS and WES, respectively) data. It is widely employed at both the Earlham Institute and WellSpan for processing raw WES/WGS data into clinically interpretable variant calls. At Earlham, it supports research integrating genomic, transcriptomic, and epigenomic data to reconstruct regulatory networks and annotate noncoding functional elements, particularly those involved in cardiac regulation. At WellSpan, GATK is a key component of The Gene Health Project, which enables individuals to voluntarily submit DNA samples for sequencing and analysis to assess their inherited risk for heart disease and common cancer syndromes. We focused on the pre-processing stage of the GATK pipeline as it is the most computationally intensive part of genome analysis in both the aforementioned use cases.

We provide a brief overview of the pipeline for setting the context here. A sequencing experiment produces a large volume of reads that must be aligned to a reference genome. The resulting BAM files contain mapped reads that require further processing to reduce false positive variant calls. The pre-processing begins by grouping reads by read group, library, and genomic coordinates. Reads are then sorted by chromosome and start position. For each set of reads with identical alignment positions, the read with the highest base quality is retained, and duplicates are marked in the SAM/BAM file header to be ignored during variant calling. Next, base quality score recalibration (BQSR) is performed. This step corrects systematic errors introduced by the sequencer when assigning quality scores to individual bases. It works by comparing observed mismatches to a set of known variant sites, learning the conditions (e.g., machine cycle, read group, sequence context) under which inaccuracies occur. The recalibration process is carried out in two stages:

1. BaseRecalibrator: Builds a statistical model using known variant sites (e.g., from dbSNP) and computes covariates that influence quality score accuracy. It then estimates empirical error rates and compares them with the reported quality scores.
2. ApplyBQSR: Applies the recalibration model to adjust the quality scores of the sequencing reads accordingly.

3 Code Re-writing

In the period M1—M9, we developed a single-threaded reimplementation of GATK in C++ and compared it against the stock GATK based on Java. We constructed a ground truth dataset to benchmark our SYCL-GAL implementation against GATK. Short reads were simulated from chromosome 14 using the error profile of the Illumina HiSeq 2500 platform, generating 12 GB of raw sequencing data in FASTQ format. The dataset included 10 variants—both coding and non-coding—one of which was a clinically validated and associated to cardiovascular diseases.

Using this dataset, we delivered a demo at the first NextGen GA meeting, where we demonstrated a 3.5-fold performance improvement (Figure 1). Notably, our implementation successfully identified all simulated variants.

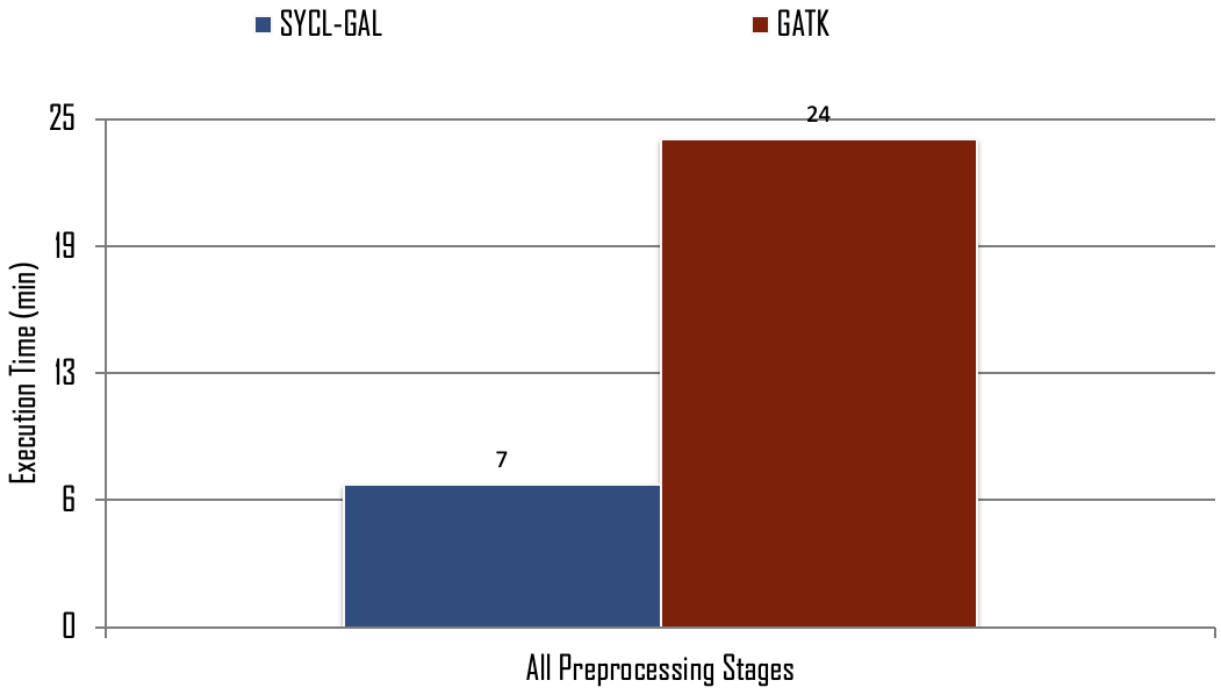

Figure 1. Comparison of single-threaded SYCL-GAL vs GATK

4 Parallelisation

In the period M9—M18, we developed a parallel, multithreaded version of SYCL-GAL. First, we optimized input/output operations. While GATK reads from and writes to disk at each processing stage, SYCL-GAL retains data in memory throughout the entire pipeline, significantly reducing I/O overhead. We also optimized the internal data structures by adopting a structure of arrays (SoA) with a columnar layout. This approach improves CPU cache efficiency and allows for fast, selective access to relevant columns at each processing stage. To support this layout, we implemented a two-stage sorting strategy: first, generating a sorted index based on specific fields (e.g., genomic coordinates), and then rearranging each data field according to this index.

In the duplicate marking step, SYCL-GAL uses hash tables to track original sequences and their duplicates based on a quality score metric. Sequences are distributed across multiple threads. Each thread accesses the hash table to compare the current sequence with the "origin"—the highest-quality duplicate. If a higher-quality sequence is found, the origin is atomically updated.

Our SYCL-GAL recalibration step follows the principles implemented in GATK. It computes auxiliary arrays (covariates) such as nucleotide context, cycle position, and read group information, and build a model, for each base, updating a recalibration table indexed by covariate combinations, recording observed errors and their frequency. Sequences are distributed to multiple threads, and each thread updates its local copy of the recalibration table. Finally, local tables are merged in a global table.

During the "Apply Recalibration" phase, base quality scores are adjusted using the computed recalibration tables. Sequences are again processed in parallel, with each thread applying corrections independently to its assigned data.

In the second NextGen GA meeting, we presented a demo where we compared our multithreaded SYCL-GAL implementation (using 64 CPU threads) to GATK and our single-threaded version. We showed a 7-fold performance improvement over GATK (Figure 2) and a 4-fold speed-up over our own single-threaded version (Figure 2). Figure 3 shows a breakdown of processing time and I/O, and as can been our multithreaded version of SYCL-GAL very effectively reduces processing time compared to the single-threaded version on the dataset described earlier, leaving I/O as the overhead.

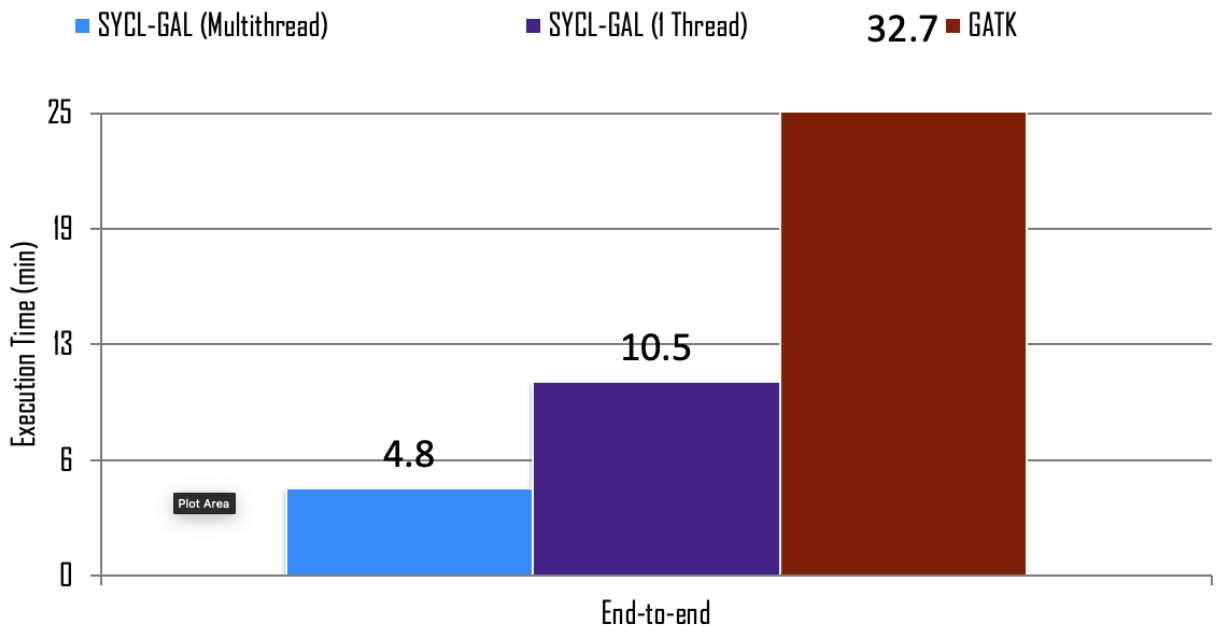

Figure 2. Comparison of single and multi-threaded SYCL-GAL vs GATK

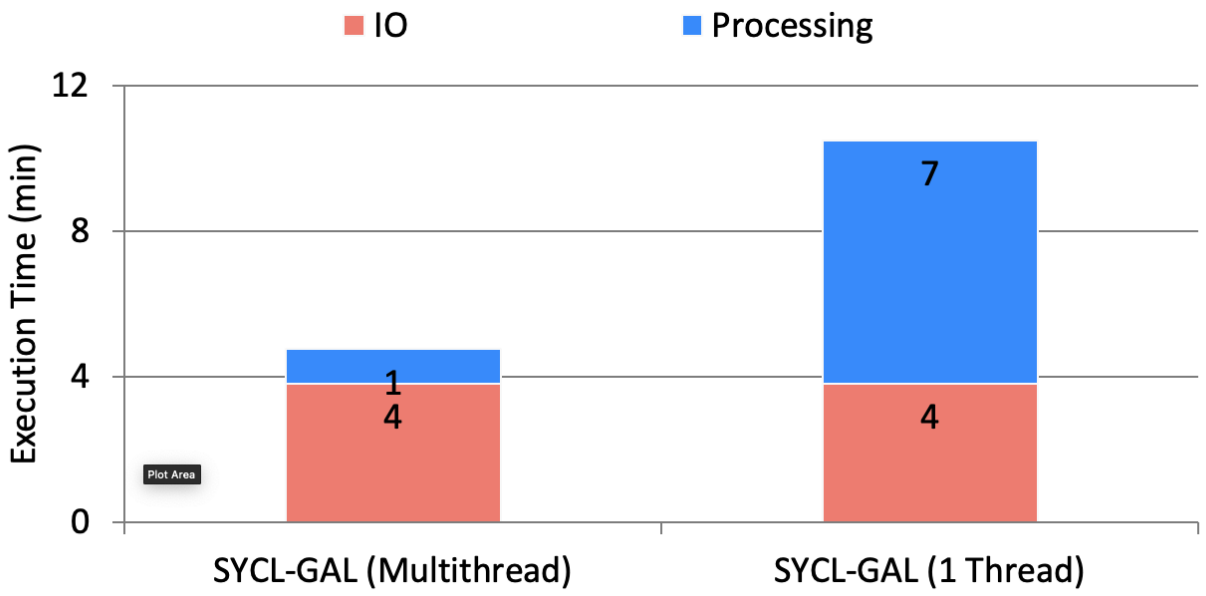

Figure 3. Breakdown of I/O vs processing time in SYCL-GAL

Figure 4 compares SYCL-GAL with Parabricks, the commercial GPU-accelerated solution from NVIDIA. Although our current CPU-based implementation is still 5 times slower than the GPU-accelerated Parabricks pipeline (Figure 4), we plan to integrate GPU acceleration into SYCL-GAL in the coming months to bridge this gap.

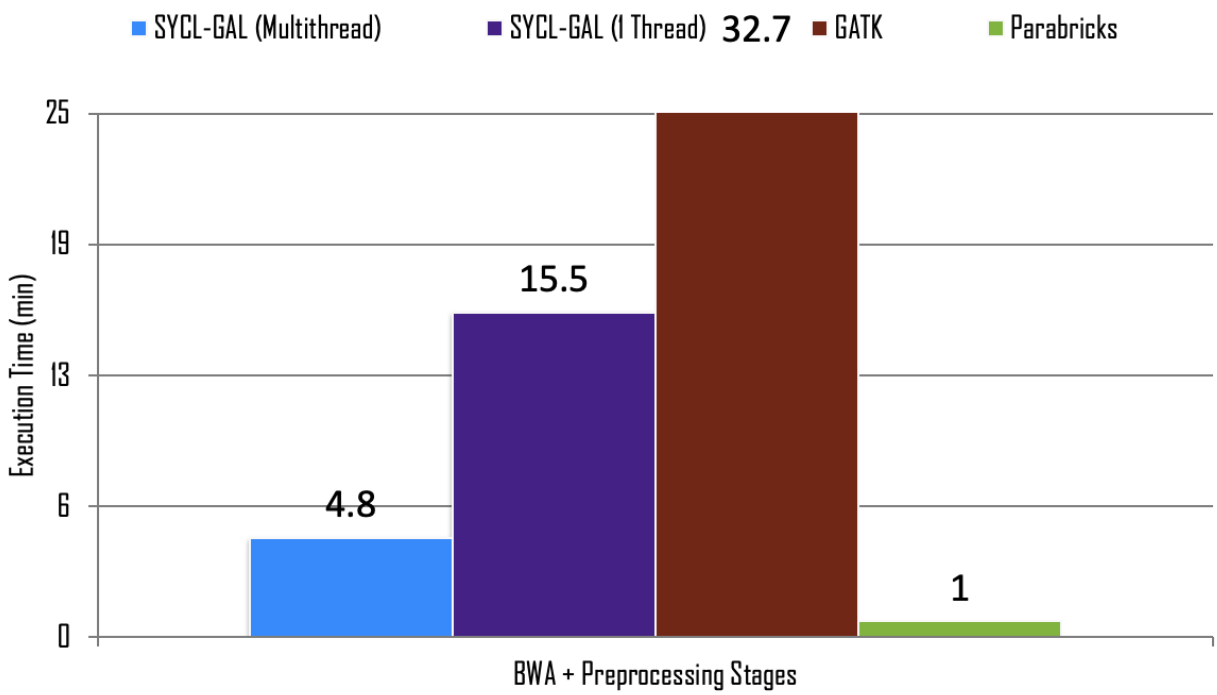

Figure 4. Comparison of SYCL-GAL with NVIDIA Parabricks

5 Conclusions and next steps

Our work demonstrates that SYCL-GAL, a re-implementation of the GATK secondary analysis pipeline, achieves significant performance improvements through CPU-based parallelization and memory-efficient data handling. By optimizing I/O operations, adopting a structure-of-arrays layout, implementing two-stage sorting, and parallelizing key steps such as duplicate marking and base quality score recalibration, we achieved a 7-fold speed-up over the original GATK and our single-threaded implementation. Importantly, SYCL-GAL maintained full accuracy, correctly identifying all simulated variants, including a clinically validated one.

While the multithreaded CPU implementation still lags behind GPU-accelerated solutions like NVIDIA Parabricks by a factor of five, it offers a flexible and scalable foundation for further acceleration. In the coming months, we will extend SYCL-GAL with GPU support to close the remaining performance gap, enabling faster, cost-effective secondary analysis for large-scale sequencing projects across a broader range of hardware platforms.