Comparison of CPU and Parabricks GPU Enabled Bioinformatics Software for High Throughput Clinical Genomic Applications

Stefano Rosati
Marquette University

COMPARISON OF CPU AND PARABRICKS GPU ENABLED BIOINFORMATICS SOFTWARE FOR HIGH THROUGHPUT CLINICAL GENOMIC APPLICATIONS

by
Stefano Rosati

A Thesis submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Bioinformatics

December, 2020

ABSTRACT

COMPARISON OF CPU AND PARABRICKS GPU ENABLED BIOINFORMATICS SOFTWARE FOR HIGH THROUGHPUT CLINICAL GENOMIC APPLICATIONS

Stefano Rosati, B.A.

Marquette University, 2020

In recent years, high performance computing (HPC) has begun to revolutionize the architecture of software and servers to meet the ever-increasing demand for speed & efficiency. One of the ways this change is manifesting is the adoption of graphics processor units (GPUs). Used correctly, GPUS can increase throughput and decrease compute time for certain computational problems. Bioinformatics, an HPC dependent discipline, is no exception. As bioinformatics continues advance clinical care by sequencing patient's DNA and RNA for diagnosis of diseases, there is an ever-increasing demand for faster data processing to improve clinical sequencing turnaround time.

Parabricks, a GPU enabled bioinformatics software is one of the leaders in 'lifting over' common CPU bioinformatics tools to GPU architectures. In the present study, bioinformatics pipelines built with Parabricks GPU enabled software are compared with standard CPU bioinformatics software. Pipeline results and run performance comparisons are performed to show the impact this technology change can have for a medium sized computational cluster.

The present study finds that Parabricks' GPU workflows show a massive increase in overall efficiency by cutting overall run time by roughly 21x, cutting overall computational hours needed by 650x. Parabricks GPU workflows show a 99.5% variant call concordance rate when compared to clinically validated CPU workflows. Substitution of Parabricks GPU alignment into a clinically validated CPU based pipeline reduces the number of compute hours from 836 hours to 727 hours and returns the same results, showing CPU and GPU's can be used together to reduce pipeline turnaround time & compute resource burden. Overall, integration of GPUs into bioinformatic pipelines leads to massive reduction of turnaround time, reduction of computation times, and increased throughput, with little to no sacrifice in overall output quality. The findings of this study show GPU based bioinformatic workflows, like Parabricks, could greatly improve whole genome sequencing accessibility for clinical use by reduction of testing turnaround time.

ACKNOWLEDGMENTS

Stefano Rosati, B.A.

I would like to thank my family for their constant support. The GSPMC for allowing for testing and computation to be done using their resources. Dr. Serdar Bozdag for his assistance navigating my entire journey through my Master's degree. Dr. Gunter Scharer for his ceaseless support of my work, personal and academic life. Dr. Sam Nie, for her flexibility, support and friendship through this process. Brian Gizelar for lending me the idea to use Parabricks, as well has his constant technical support. I also would like to acknowledge the Marquette Graduate School and Marquette University administration for its guidance and support as I navigated the life of a part time student.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Background

The fields of bioinformatics and clinical genomics have grown rapidly with the rise of next generation sequencing and the subsequent increase in publicly available data and decrease in the cost to generate genomic data. Despite the growth of the clinical genomics, clinical sequencing still faces challenges in becoming a routine diagnostic test, causing it to remain a last-ditch effort to end a diagnostic odyssey. Of the major challenges facing whole genome sequencing (WGS), the high cost and long turnaround time (TAT), the time from test order through results reporting, remain the largest hurdles to making whole genome sequencing (WGS) a staple of diagnostic testing (Manolio, 2017). While advances in sequencing technology and testing availability have improved clinical WGS TAT, bioinformatic computation and variant analysis remain a challenge due to the volume of data and amount of hands on time required by analysts (Miller, 2015). Within this study, a new graphics processor unit (GPUs) based bioinformatics toolset called Parabricks is assessed against standard bioinformatics pipelines using central processing units (CPUs), showing drastic reduction of TAT with no sacrifice in clinical efficacy.

Clinical Genomics

Since the start of the Human Genome Project, integration of computer science into genetic sequencing has played a key role in the growth of genomics for research and clinical applications (Lander, 2001), (Hood, 2003). As computer science and data

analysis have become more intertwined, the cost of sequencing has reduced drastically. Estimates published in 2018 show clinical whole exome sequencing (WES) ranges from $555 to $5,169 and clinical whole genome sequencing (WGS) from $1,906 to $24,810 (Schwarze, 2018). While current WES and WGS costs are still prohibitive for most patients, the current price is a drastic reduction from the 2.7 billion dollar price tag of the first human genome in 2003 (National Institute of Health, n.d.). The decrease in sequencing cost is driven by technological innovations allowing for high throughput sequencing of many samples in parallel called Next Generation Sequencing (NGS) (Schwarze, 2018). As the cost of sequencing continues decreases, NGS has continually showed its ability to transforming the diagnostic methods for finding causes of both rare & common mendelian diseases, as well as providing accurate diagnosis and targeted treatment of cancer (Cirulli, 2010) (Willig, 2015).

Despite the price decrease of WES and WGS testing in recent years, the world of clinical genomics still faces challenges. The largest challenges remain turnaround time (TAT), variant interpretation and data management (Rossen, 2018) (Meienberg, 2016). A single patient's WGS data can result in hundreds of gigabytes of data and can take thousands of hours of server time (Muir, 2016). In recent years, the growth of cloud computing and storage has decreased the cost of maintaining servers and have made it easier to scale up workflows, but has not done much to increase the use of WGS over using smaller exome and genome panels (Muir, 2016). At present, it remains much more effective for clinicians to order smaller gene panels and WES than WGS due to the lower cost and faster TAT (Muir, 2016). Smaller gene panels are attractive to clinicians and

patients due to their lower cost, as they require less sequencing, less computation and less hands-on analytical time.

Despite the drawbacks of high data volume, long compute time, and costly hands variant curation requirements, there is little disagreement that WGS is a highly effective method for clinical diagnostics. Many clinicians argue that WGS should be a front line defense for neonatal crises and should be utilized in the management of acute medical care (Miller, 2015) (Saunders, 2012) (Manolio, 2017). WGS has been shown to help in these scenarios by assisting in diagnosing rare and new diseases, diagnosing cases with atypical presentation and can help in cases where standard treatments are ineffective (Miller, 2015). WGS findings additionally assist clinicians in managing and treating diseases by giving insights into a disease's etiology, offering clinicians insights into treatment as opposed to long term management of symptoms (Clark, 2019). WGS sequencing also has the ability to show clinicians when irremediable damage is done to the genome, allowing them to begin palliative care knowing they have done all they can for their patients without prolonging suffering (Willig, 2015).

WGS additionally provides methodological benefit by superior data to target sequencing methodologies such sequencing by exome capture or PCR amplification. WGS' indiscriminate method of genome surveillance and variant detection can lead to detection of previously unknown pathogenic disease origins (Lionel, 2018). Whole genome sequencing also has better resolution for calling copy number variants (CNVs), large genomic deletions or insertions, by increasing the probability of sequencing over

breakpoints and by giving better resolution of large genomic events (Lionel, 2018). By contrast, exome and targeted panels have difficulty detecting CNVs, due to the finite targets, and limited resolution (Zhao, 2013). Perhaps counterintuitively, a negative result from a WGS test is more significant result than a negative targeted panel. This means that a negative result from WGS testing is less likely to leave clinicians and patients wondering if they need to order a more comprehensive or advanced test.

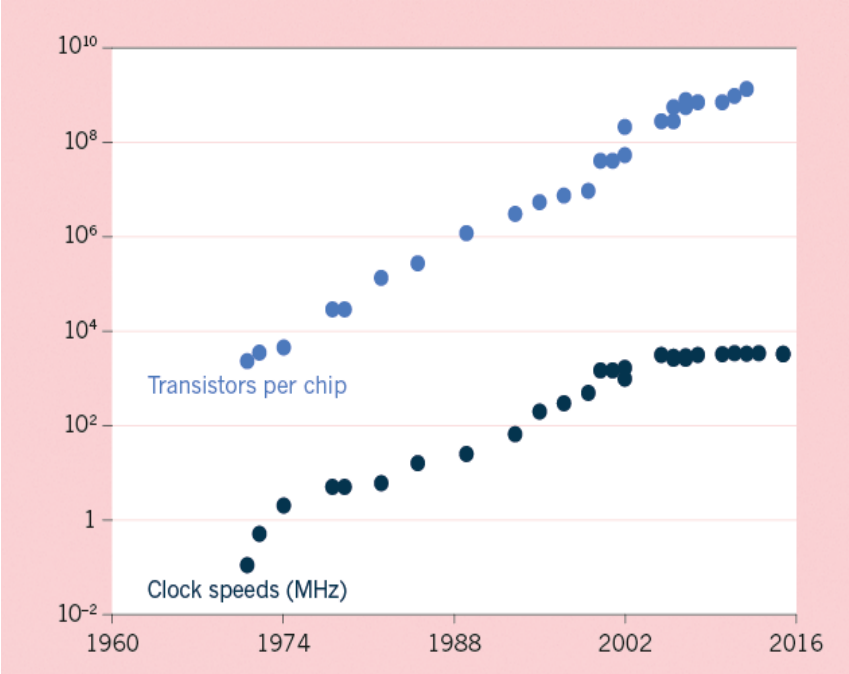
Another scenario in which WGS sequencing has utility is neonatal crisis, where any of an estimated 500-1000 different genes can drive newborns to rapidly deteriorate (Kingsmore, 2012). Neonatal crises account for around 20% of newborn deaths in the US and up to 18% of newborn hospitalizations (Kingsmore, 2012). Many these syndromes and metabolic conditions are reversible and preventable if detected early and treatment is administered within a reasonable amount of time. Two such examples of preventable & manageable diseases include phenylketonuria (PKU) and congenital hypothyroidism which effect 1 in 10,000 and 1 in 2,000 newborns respectively (Kingsmore, 2012). While the cost of WGS testing might seem cost prohibitive for routine use in diagnoses of rare diseases, the cost of testing is much more reasonable when compared to a daily price tag of $3,500+ in the Neonatal Intensive Care Unit (NICU) (Muraskas & Parsi, 2008). However, current WGS turnaround times of 1-2 months remain woefully insufficient to make an impact in many cases (Thiffault, 2019). The speed of WGS sequencing and analysis continues to prevent the power of the method from being brought into everyday clinical use.

Computational Challenges

Central processing units (CPUs) are the standard calculator that perform computations within a computer. Throughout the rise of computers, the driving force increasing computer performance and software speed was the advancement of CPU speed. This concept has termed “Moore’s Law”, named after the author of the landmark 1965 paper, posits, “The complexity for minimum component costs has increased at a rate of roughly a factor of two per year” (Moore, 1998). This phrase was come to be understood that computational power of standard CPUs would double every 12-18 months and has been remarkably accurate over the last half century.

More recently, Moore’s law has run out of runway. While transistors per chip continues to double every 12-18 months, speeds of processors have reached a plateau (Waldrop 2016) (Figure 1). The driver of this plateau is that CPU hardware is approaching fundamental physical limitations, in which speeds cannot improve the speed without employing super cooling or involving massive power consumption (Markov, 2014). The consequences of this phenomenon are far reaching. Previously, if a developer wrote a piece of software, all they would need to do to make it go faster is wait for a faster generation of processers to come out. Now, to achieve greater performance developers need to get more creative in their coding or must look to new computer architectures to increase efficiency.

Figure 1: A plot showing the relationship between clock speed and transistors per chip over the last 50 years. (Waldrop, 2016)

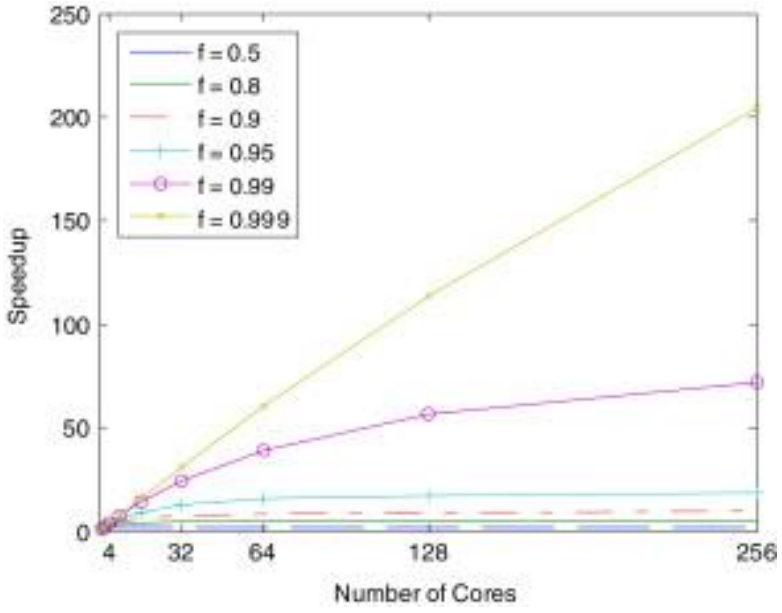


One such architectural solution is the multi-core processor, in which multiple processors work in sync to accomplish tasks. Provided the software used can operate with multiple CPUs, this improvement allows for large tasks to be spread across many CPUs to accomplish the task sooner. While multicore processing can increase the number of computations that can be done in a given amount of time, it has some notable drawbacks. One of the most important predictors of a computational task's speed increase is parallelizability, termed "f".

Each unique combination of computational task, operating system and hardware combine to create a unique f value. In general, each additional CPU assigned to a task increases the speed but depending on the parallelizability of the task (f), there is a

diminishing return in overall CPU efficiency (Figure 2). This diminishing return on CPU resource investment is known as Ahmdal's law (Sun, 2010). This means that efficiency and fold speed up of adding more CPUs to speed up a computation is gated by f. As f decreases, so does the return on investment for each CPU added to a computation. As bioinformaticians and software engineers look to increase the speed of WGS pipelines, they are forced to re-design how their code functions, break tasks into many parts, or look to other computational architecture in order to increase the parallelizability.

Figure 2: A representation of the relationship between the parallelizability (f) of a computational job, the number of CPU cores assigned to it, and the fold speedup of the job's completion (Sun, 2010)



Graphics processor units (GPUs) represent another of the solution to HPC throughput. GPUs initial purpose were specifically for high resolution screens- where computations must be performed to render millions of pixels hundreds of times per

second. GPUs are rapidly being adopted into HPC workflows as they can outperform CPUs for tasks that require many simple computations at once (Nickolls, 2010). By using code specifically designed for GPUs, GPU architectures can work in synergy with CPUs to massively speed up computational jobs. While GPUs can represent a large performance upgrade, the drawback is increased difficulty in code design and great difficulty involved in troubleshooting.

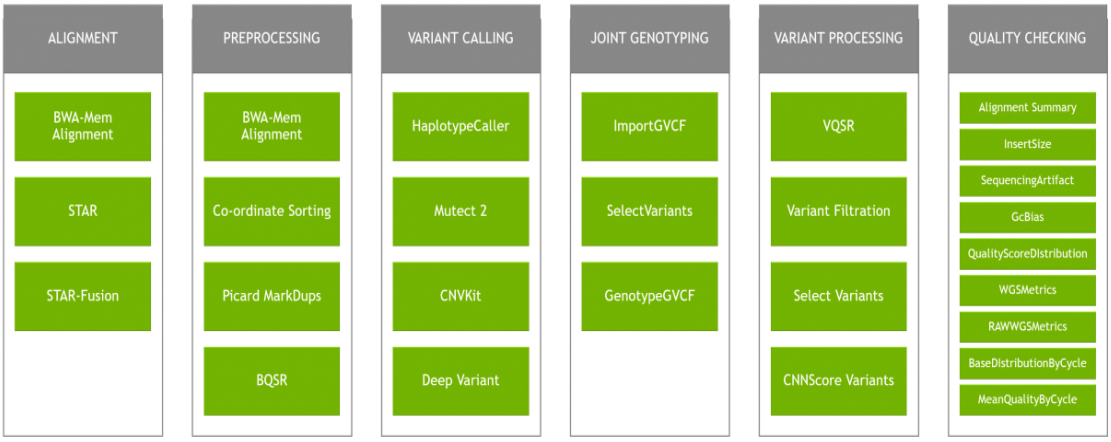
Bioinformatics as a discipline is extremely familiar with high complexity HPC problems. For example, in alignment of a single sequencing 'read' consisting of a string of 200-300 As, Ts, Cs, and Gs to the human genome, must compute the 'optimal' placement of the read into to a genome 3.2 billion bases in length. This matrix multiplication must be performed one or more times for each sequencing read in a genome. A standard clinical WGS sequencing can produce upwards of 500 million reads. Meaning this N x M computation could be performed 500 million times to create a single base alignment map file (BAM file). Even divided across many different CPU processors, this computation can take hours or days to complete. Fortunately, genome alignments matrix multiplication is what many in the computer science industry call a "ridiculously parallel" problem, meaning no single computation is dependent on any other. This makes alignment and many bioinformatic tasks perfectly suited to GPU workflows. Until now, one of the greatest drawbacks to implementing GPU workflows is the difficulty of troubleshooting errors when they arise, and the difficulty of implementing GPU workflows into pipelines.

GATK & Parabricks Bioinformatics Tool Suites

The Genome Analysis Tool Kit (GATK) is a suite of open sourced highly versatile Java based bioinformatics tools from the Broad Institute is an (Van der Auwera, 2013). GATK has tools ranging from raw read trimming to *de novo* genome assembly to alignment of raw reads to reference genomes to somatic & germline variant calling (Rimmer, 2014) (Van der Auwera, 2013). GATK's wide user base and vast documentation makes is a favorite of bioinformaticians. While extremely reliable, GATK's major drawback is currently that it is CPU based and its distributed SPARK architecture has been for development use only since 2017.

Parabricks, now a product of NVIDIA, is a commercially available tool which converts GATK based functions and algorithms their native CPU architecture to a GPU architecture (Figure 3). Parabricks functions by keeping the underlying GATK tools the same, but adapts the most critical & parallelizable algorithms over to GPU enabled CUDA code (NVIDIA (Parabricks), 2020) (NVIDIA (CUDA), 2020). The Parabricks software

Figure 3: Currently available Bioinformatics tools within the Parabricks software suite. (Parabricks, 2020)



is specifically written for the NVIDIA DGX-1, a single unit that contains 8 separate GPUs. Parabricks boasts that their software can cut GATK pipeline turnaround times (TAT) by 40-60 times (NVIDIA (Parabricks), 2020). If true, this reduction in TAT poses a breakthrough in bioinformatics that has the ability to increase WGS throughput, cut WGS TAT and have a positive impact for patients and clinicians alike.

Study Objective

The goal of this study is to test Parabricks GPU enabled bioinformatic workflows and tools against clinically validated CPU bioinformatic workflows for efficacy of clinical whole genome sequencing (WGS). Evaluations for each pipeline include overall bioinformatic workflow turnaround time (TAT), genome alignment TAT and performance assessment of variant calling on National Institute of Standards and Technology (NIST) Gold Standard Genome in a Bottle sample NA12878. Parabricks claims to reduce pipeline TAT by 40x to 60x over standard GATK pipelines with no sacrifice in output quality. If their claims are true, Parabricks has ability to revolutionize whole genome sequencing by reducing WGS TAT and increasing the clinical utility of WGS testing for labs, clinicians, and most importantly patients.

METHODS & MATERIALS

Sequencing Data

The comparisons performed within this document were performed on 3 de-identified human samples from DNA. WGS-High Coverage (WGS-HC), a deeply sequenced genome sample with about 150x coverage of the genome, WGS-Normal Coverage (WGS-NC), 55x sequencing depth, and the Genome in a Bottle (GIAB) NA12878 national reference sample purified from cell line (National Institute of Standards and Technology, 2020). All samples had library preparation with Illumina's TruSeq Nano Whole Genome preparation per manufacturer specifications (Illumina, 2020). All samples were sequenced bi-directionally with 2x150 reads on the same Illumina NovaSeq using an SP flowcell. Raw read data was prepared using of BCL2Fastq2 v2.20 and was stored in fastq.gz format (Illumina, 2019).

Pipelines Tested

3 Separate pipelines were tested within this study all aligning to Genome Reference Consortium Human Build 37 (GRCh37) genome (Church, 2011). The three pipelines tested in this study were: 1) 'Clinically Validated CPU' based, Whole Genome Sequencing (WGS) pipeline based on GATK 2) 'Parabricks Rapid', a germline pipeline for WGS built only on the Parabricks tool suite, 3) 'Parabricks hybrid' pipeline, a combination of pipeline 1 and 2, in which the alignment is performed by Parabricks GPU BWA alignment algorithm. All other tasks are performed by the standard CPU Pipeline. The non-quality control pipeline features are summarized in Table 1.

Table 1: Non-Quality control steps and software performing them in CPU, Hybrid and GPU pipelines tested within this document.

Pipeline Step	CPU Pipeline	Hybrid Pipeline	Parabricks GPU pipeline
Alignment	GATK BWA-MEM	Parabricks BWA-MEM Align + MarkDuplicates	Parabricks BWA-MEM Align + MarkDuplicates
Mark Duplicates	Picard Mark Duplicates	NA	NA
Quality Score Recalibration	GATK Base Quality Score Recalibration (BQSR)	GATK Base Quality Score Recalibration (BQSR)	Parabricks BQSR
Variant Caller	GATK haplotype caller	GATK haplotype caller	Parabricks Haplotype Caller
Genotype caller	GATK genotype gVCF	GATK genotype gVCF	NA
GATK Variant Normalization	BCFtools Norm	BCFtools Norm	NA

All GATK applications used within this study were performed on GATK version 4.1.2.0 (DROAZEN, 2019). Parabricks runs were performed on v2.4.6, using GATK functions were lifted over from GATK v4.1.2.0.

Computation Environment

CPU based computations were performed on a 15 node HPC cluster consisting 340 Intel(R) Xeon(R) Gold 6154 3.00GHz CPUs, each node containing 6 TB of ddr3 memory. GPU computations are performed on a Nvidia DGX-One with 2-24 core 2x Xeon Gold 8268 CPUs, and 8 NVIDIA Tesla P100 GPUs, and 512GB DDR4-2133 + 128GB HBM2 memory. All systems used Linux Centos 7 as the operating system. All CPU and

GPU compute jobs were managed by Torque v4.2.6 (Adaptive Computing, 2013) . GPU computations were performed using CUDA v 10.1 (NVIDIA (CUDA), 2020).

Alignment Benchmarking

As genomic alignment tends to be the rate limiting step of most genomic pipelines, an initial benchmarking of Parabricks Alignment + MarkDuplicates (BWA) algorithm was performed using 2, 4 & 8 of the dedicated GPUs each in quadruplicate. During the 2 and 4 GPU Parabricks alignment + mark duplicates assessments, all DGX GPUs not under assessment were assigned alignment jobs to simulate uniform input / output volume across the entire unit. Using the same sample input data, A GATK BWA-MEM alignment + Mark Duplicates was performed in quadruplicate with 8 CPUs dedicated using GATK best practices (GATK (Best Practices), 2020). Turnaround times of 2, 4 and 8 GPU Parabricks alignments were compared to CPU alignments.

Pipeline TAT Benchmarking

To test the turnaround time of the CPU, GPU and hybrid pipelines, each pipeline was run from fastq files using sample WGS-HC (150x genome coverage), WGS-NC (55x genome coverage), aligning to GRCh37. Each pipeline was run in duplicate and the turnaround time was averaged. Turnaround time was measured from pipeline start to completion.

A second analysis was performed on the same pipeline runs to compare the Hybrid and CPU pipelines. The TAT of each step of the critical path, or longest path of interdependent steps, was used to compare the difference Parabricks GPU alignment to

GATK CPU alignment when all other steps remain the same. Job statistics were obtained from the Torque scheduler logs.

Variant Calling Performance

The variant calling output for all three pipelines were assessed using NIST GIAB sample NA12878 was used to assess variant calling performance of each pipeline. The published NIST NA12878 'High Confidence' variant call set was used to assess each pipeline's output VCF file (Zook J. M., 2019). The bed file for NA12878 high confidence variant call files was bed intersected over all pipeline VCF files (Quinlan, 2010). The VCF files for each pipeline, restricted to the high confidence regions, were then compared to the NIST published high confidence variant calls from sample NA12878 using VCFtools' vcf-compare (Danecek, 2011). The definitions of true positives, true negatives, false positives and false negatives shown in Table 2 were used to assess performance.

Table 2: Definitions of performance metrics as used to assess GPU, CPU and hybrid pipeline clinical efficacy.

Term	**Definition**
True Positive (TP)	Matching reference and alternate allele between pipeline output and published NA12878 high confidence call set
True Negative (TN)	Site within the NA12878 high confidence region bedfile, but with no variant calls in both pipeline VCF and NA12878 high confidence variant call file
False Positive (FP)	Site with variant call present in pipeline VCF output, but no matching call in the NA12878 high confidence variant call file
False Negative (FN)	Site with no call in pipeline results VCF and a variant present in NA12878 high confidence variant call
Sensitivity	TP / (TP + FN)
Specificity	TN / (FP + TN)

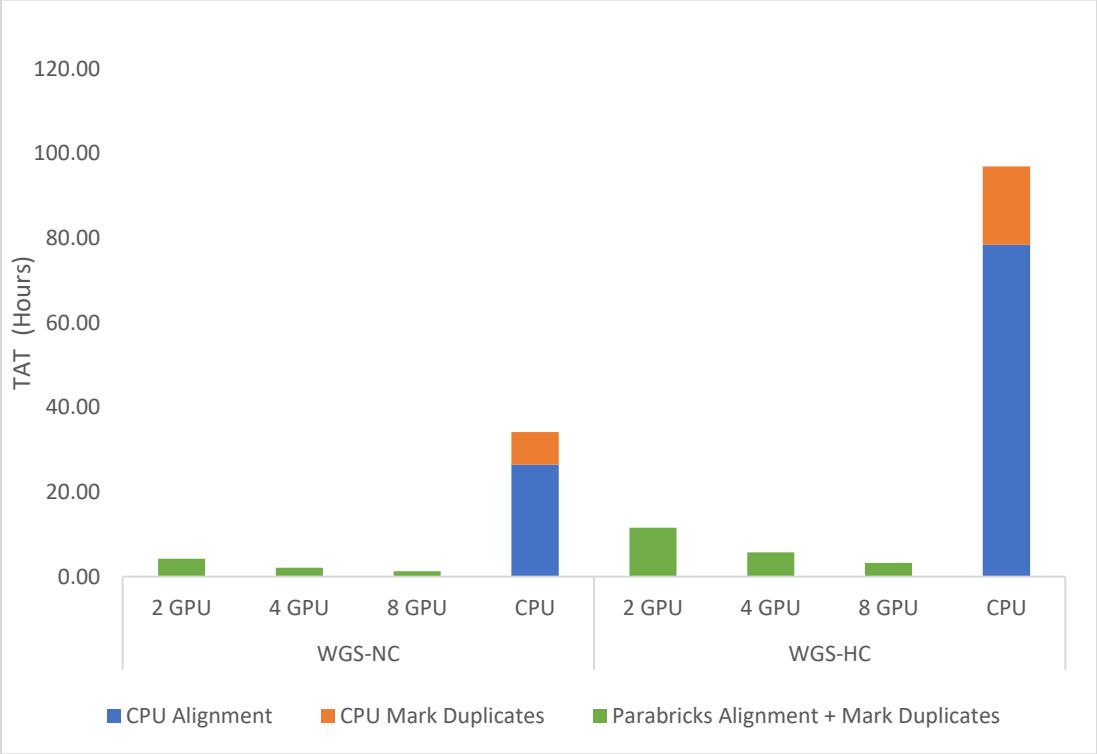
Positive predictive value (PPV)	TP / (TP + FP)
Negative Predictive value (NPV)	TN / (FN + TN)

RESULTS

CPU Alignment vs GPU Alignment

To assess the efficiency of alignment by Parabricks, the same fastq files were run on the GATK based CPU pipeline, as well as the Parabricks pipeline, using 2, 4, and 8 dedicated GPUs to align and mark duplicate reads (Figure 4). All CPU alignments were performed given 8 CPUs. All alignments on the CPU and GPU pipelines were performed using the same GATK alignment parameters.

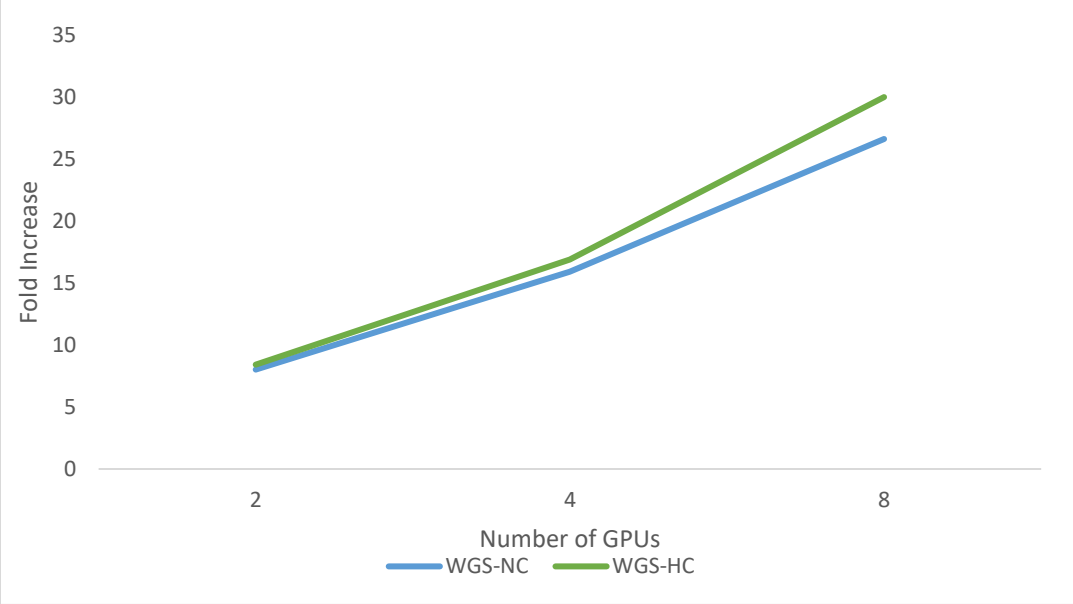
Figure 4: Alignment + Mark Duplicates comparison between Parabricks GPU alignment using 2, 4 and 8 dedicated GPUs and GATK CPU alignment with 8 CPUs allocated.



Comparing run times of the CPU and GPU architecture shows a stark difference the overall time it takes for the genomic alignment and duplication marking to complete. Running with only two dedicate GPUs, the alignment and mark duplicates together runs twice as fast as the CPU duplicate marking alone. Comparing GPU turnaround time of 150x depth WGS-High Coverage (WGS-HC) to the 55x depth WGS-Normal Coverage (WGS-NC) shows that the GPU alignment time is linear to the sample's sequencing depth

Further comparison of the CPU and GPU Alignment + Mark Duplicates speed shows increase was highly correlated to the number of GPUs dedicated to the job. Tests were run with 2, 4, 8 GPUs dedicated to the alignment job showed an 8x, 16x, 28x speed increase in alignment respectively (Figure 5). Showing that the GPU architecture does not suffer from a diminishing return when increasing allocated processing units.

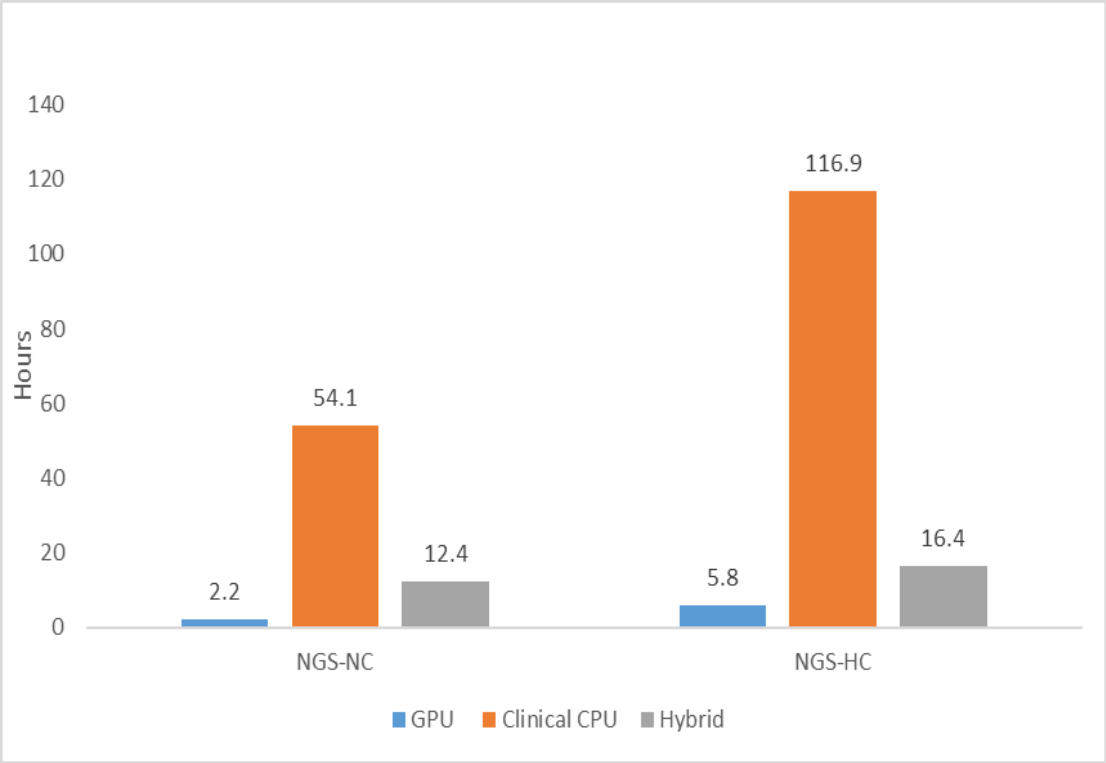
Figure 5: Fold speed increase of 2, 4, 8 GPUs dedicated to alignment of compared to standard CPU alignment of two separate WGS samples.



Turnaround Time Comparison: GPU vs CPU Pipeline

To assess full pipeline turnaround time, all 3 pipelines were run on WGS-HC and WGS-NC samples. The TAT is compared below to show TAT from job submission through completion for (Figure 6).

Figure 6: Comparison of TAT of GPU, CPU and hybrid pipeline, for NGS-HC and NGS-NC samples

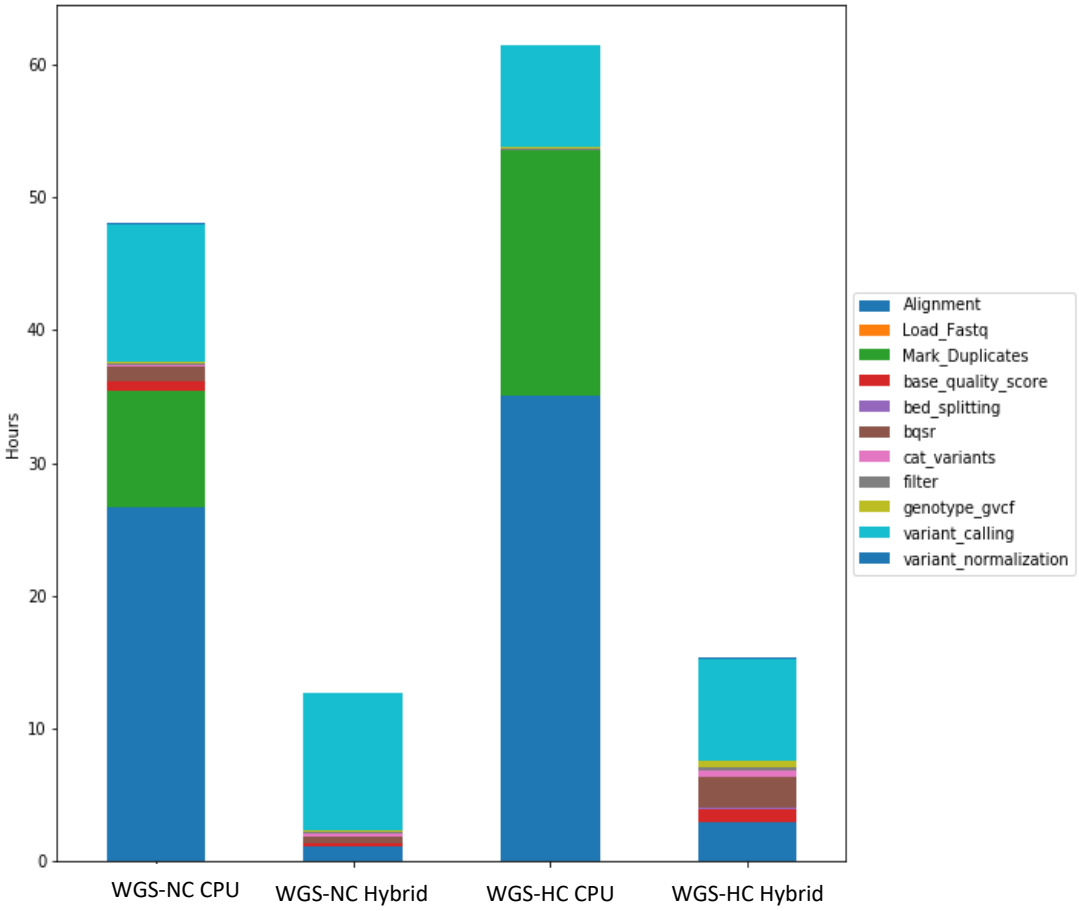


The GPU pipeline speeds up the fastq to VCF TAT by about 25x when compared to the CPU pipeline. The hybrid pipeline shows roughly a 5x increase in TAT.

Turnaround Time Comparison: Hybrid Pipeline vs CPU Pipeline

A hybrid pipeline was created by combining the Clinical CPU pipeline and Parabricks offloads alignment, mark duplicates, and bam sorting, 3 of the longest tasks in the Clinical CPU Pipeline. The two pipelines were run on the WGS-NC and WGS-HC samples. A comparison was performed of the critical path steps – or the steps that are dependent on a prior step before they can start (Figure 7).

Figure 7: Compute hours of the longest critical steps between CPU and Hybrid pipelines, showing TAT of each step in the pipeline's critical step for the WGS-HC and WGS-NC samples.



For the CPU pipeline, the alignment and mark duplicates took more than 75% of the compute time required to complete the critical steps of the pipeline. During this time, the resources dedicated to those tasks are both unable to be used by other tasks, and other essential processes in the pipeline cannot proceed because they are dependent on the aligned, sorted bam file. In the Hybrid pipeline, the alignment and duplicates complete 25-27 fold faster, allowing for all of the downstream tasks dependent on the completion of alignment and mark duplicate steps to complete to start hours sooner. The hybrid pipeline completed overall 5 times faster than the CPU pipeline did alone.

Pipeline Variant Calling Efficacy

NA12878, the NIST reference sample, was run from fastq on the clinically validated CPU pipeline, GPU Rapid pipeline and the Hybrid pipeline. The resulting variant call files were bed intersected using Bedtools with the NIST published high confidence (HC) regions bed file, where many sequencing technologies were employed to create a high confidence consensus VCF file for SNPs and small INDELs (Zook J. M., 2019) (Quinlan, 2010). The high confidence region consists of 2,575,632,881 bases. Variants within each pipeline's 'high confidence' VCF file was compared to the published high confidence variant call set using VCFTools' VCF-compare function (Danecek, 2011). Variant calls from the 3 pipelines were filtered using BCFtools to only include variants greater than 8x depth and QUAL scores greater than 20, no other filters were used in order to compare all three pipelines on a level playing field (Li, 2011) (Table 3).

Table 3: Variant comparison between NIST NA12878 High Confidence variants and assessed pipeline VCF outputs in the same regions.

Pipeline VCF file	True Positives	False Positives	False Negatives	True Negatives
Clinical CPU pipeline	3,577,524	66,604	119,070	2,571,869,683
Hybrid (GPU align only)	3,577,532	67,597	118,950	2,571,868,802
Rapid (GPU)	3,675,949	67,419	86,782	2,571,802,731

Variant calling comparisons from all three pipelines to the NIST HC call set showed 95.25% variant concordance for the Clinical CPU pipeline & Hybrid pipelines, and 97.88% concordance for the GPU pipeline (Table 4). The hybrid pipeline and CPU pipeline only had 8 different variants calls from one another, all at sites with less than eight total reads. While the Rapid GPU pipeline had 98,417 and 98,425 more calls matching the NIST High confidence variant call set, than the Clinical CPU and Hybrid CPU pipeline, respectively. Performance metrics were calculated for the three pipelines to show the overall efficacy of variant calling between the three pipelines (Table 4).

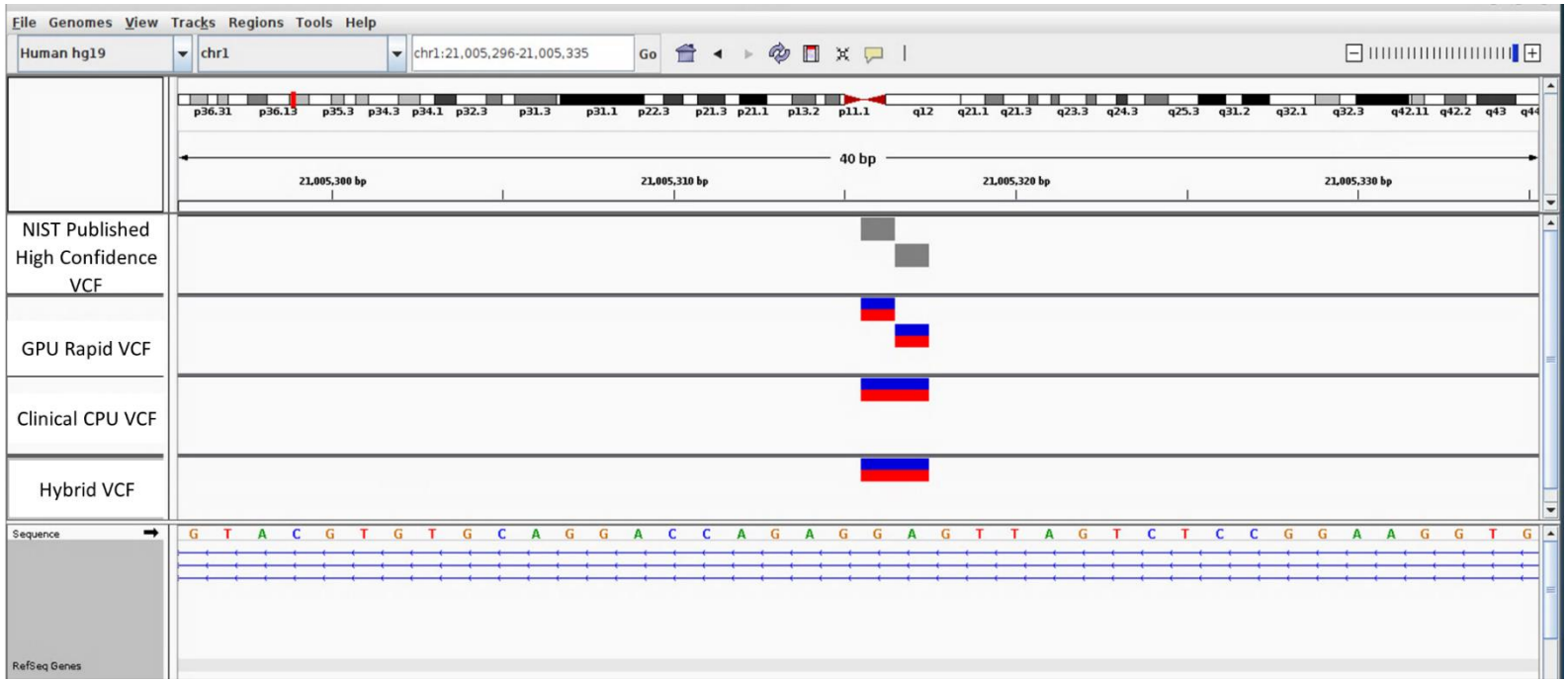
Table 4: Performance metrics showing the sensitivity, specificity, positive predictive values and negative predictive values of all three pipelines as compared to the NIST High Confidence Variant set.

Pipeline	Sensitivity	Specificity	Positive Predictive Value	Negative Predictive Value
Clinical CPU pipeline	0.96779	0.99997	0.98172	0.99995
Hybrid (GPU align only)	0.96782	0.99997	0.98146	0.99995
Rapid (GPU)	0.97694	0.99997	0.98199	0.99997

All three pipelines return similar performance results, however, the GPU based Rapid pipeline displayed a slightly higher sensitivity. All three pipelines show high quality sensitivity and positive predictive value and superb specificity and negative predictive value (Table 4).

An analysis of the discordant variants was performed to understand the differences between each pipeline output. VCF output differences between the GPU pipeline and the Hybrid pipeline were largely due to differences in their handling of multiple nucleotide polymorphism calling (MNPs) (Figure 8). Functionally, the protein product of the calls between all 4 files shown in figure 8 remain the same as the variants are call as in-phase, meaning they occur on the same strand of DNA. The difference is in the results format of the MNPs vs the SNPs is the format of the quality metrics produce by the variant caller. SNPs each receive their own quality assessment, while MNPs have their quality data merged for the combined variant.

Figure 8: IGV Screenshot showing comparisons of NIST High Confidence Callset, VCF, GPU VCF, CPU VCF and Hybrid VCF files (top to bottom), showing the same nucleotide changes between sites, but with the Clinical and Hybrid pipelines joining the calls as a multiple nucleotide polymorphism, while the NIST and GPU pipelines call two separate MNPs.



DISCUSSION

Turnaround Time and Efficiency

Implementation of Parabricks GPU enabled bioinformatic tools displays a clear advantage for process turnaround time. Two samples were selected for test runs: an average depth genome WGS-NC (~55x), and WGS-HC (~150x). The genomes were run on the Parabricks germline pipeline with 8 GPUs dedicated and CPU pipeline with 8 CPUs dedicated to assessing the turnaround time of the two alignments. The results were then used to create estimations below were made for a standard 35x coverage genome. 35x TAT assumptions were made under the assumptions that CPU and GPU compute hours are linear to the sequencing depth, and compute hours in Table 5 are available.

Table 5: Estimated CPU/GPU core hours available per year in the cluster described in this experiment.

CPU core hours avail per year	2,978,400
GPU core hours avail per year	70,080

This analysis estimates that a 35x coverage genome sample will require a total of 836 CPU hours per genome. Using the estimations in table 5, this means the CPU cluster described in this document can run roughly 3,562 genomes annually (Table 6).

Table 6: Estimated number of genomes able to be processed annually by the CPU pipeline alone on the cluster described in this document.

Assay	**CPU Hour estimation** (hours/sample)	**Maximum throughput per year**
WGS	836	3,562

Based on the results of this experiment implementation of the hybrid pipeline, hence offloading the alignment to the GPU, would offload roughly 101.8 CPU hours per whole genome pipeline run. Using the core hour estimations described in Table 5, we can estimate how many more genomes can be run annually before saturating the system (Table 7). Simply adding Parabricks GPU alignment and Mark Duplicates into CPU pipeline, the cluster described in this document can process 456 (11.3%) more genomes through the pipeline, with faster turnaround time.

Table 7: An estimate of the number of 35x depth genomes that could be run annually on the CPU and GPU cluster described within this document via the Hybrid pipeline.

CPU Hour estimation (Alignment excluded)	**GPU Hour estimation** (Alignment + Mark duplicates only)	**Maximum CPU processing** (all pipeline steps excluding alignment, mark duplicates, BQSR)	**Maximum GPU processing** (alignment + Mark duplicates only)	**Maximum throughput per year**
727.9	4.36	4,092	4,018	4,018

Running the Parabricks Rapid pipeline alone, assuming 35x coverage and 100% up time, a single DGX GPU system could process 2,037 genomes in a year (Table 8).

Table 8: An estimate of the number of 35x whole genome pipeline runs that could be run annually on the CPU and GPU cluster described within this document via the Parabricks alone.

GPU Core Hour estimation (hours)	**Maximum WGS throughput per year**
17.20	2,037

It is worth noting that the CPU and hybrid pipelines have more quality control steps built in than the GPU pipeline. Steps including GATK DepthOfCoverage, calculation of coverage for all gene regions and variant quality checks are extremely important steps for quality control and quality assurance in any clinical workflow. These steps ensure that all regions of the genome are accurately represented within the sample, helping analysts ensure that lack of variation is not confounded with lack of sequencing data. The power of the hybrid pipeline is the shortened of alignment, the longest step in the pipeline, combined with the added QC performed by the pipeline. Shortening required alignment time allows for all other dependent tasks start processing sooner, allowing for more efficient distribution of downstream jobs throughout the rest of the pipeline. In contrast, the clinical CPU workflow has a major bottleneck at the alignment step. While the CPU alignment is occurring, all other downstream jobs are waiting for alignment to complete, leading to an imbalance of CPU demand towards the end of the pipeline.

Given the speed of the GPU pipeline, another potential method for reducing overall WGS TAT is using Parabricks for both alignment and variant calling, then using

slower CPU quality assessments while the clinical variant assessment of the resulting VCF begins. The prospect of this is extremely attractive, as it would allow a clinical to go from DNA to analysis within two days. Allowing for one day on the sequencer, 2-5 hours for Parabricks to generate a VCF, and finally uploading to analysis software. Meanwhile, QC steps performed on the BAM and VCF can occur simultaneously. If any quality issues such as low coverage or low quality are found, clinical variant analysis can be halted, and sequencing can be repeated to improve quality.

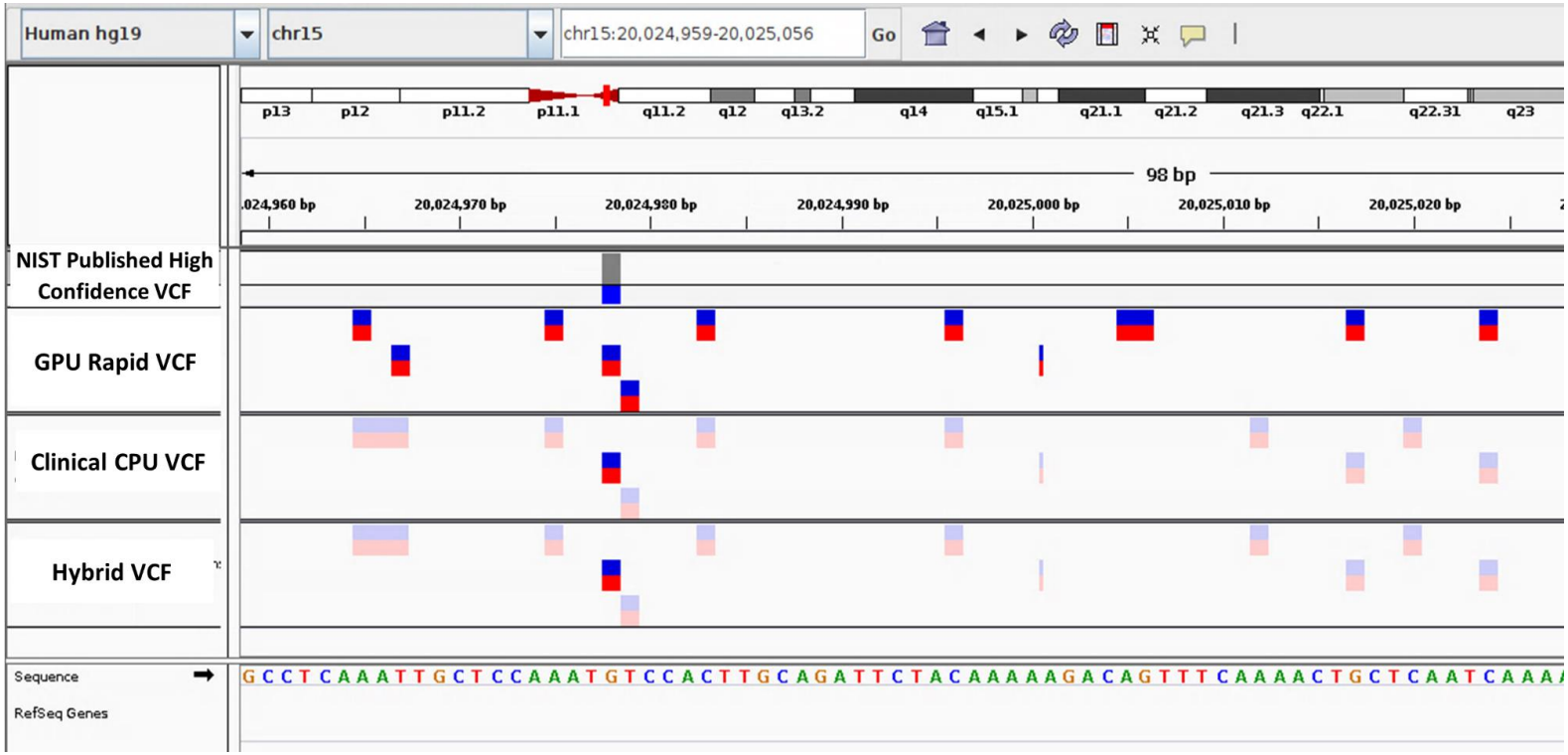
Variant Calling Efficacy

A bioinformatic pipeline is only as useful as it is accurate. To ensure the pipelines tested herein produce accurate variant calls, whole genome sequencing of the NIST Gold Standard reference sample NA12878, was run on each pipeline (Zook J. M., 2019). The results for all pipelines showed a high degree of efficacy for use in a clinical setting. In comparison with the gold standard within the high confidence regions, the Clinically validated CPU pipeline showed the worst sensitivity and positive predictive value (PPV), at 96.78% and 98.17%, respectively. The Hybrid pipeline showed slightly better sensitivity and PPV, at 96.78 and 98.15%, respectively. The rapid pipeline showed by far the most robust sensitivity and PPV, at 97.69% and 98.19%, respectively.

While all three pipelines show impressive results, it is notable that raw variant concordance can be slightly misleading. As mentioned earlier, the Rapid pipeline has very little built in QC. Figure 9 shows a single variant within a homopolymer repeat of the NIST High-Confidence Variant call set. The NIST high confidence variant set only

reports a single mutation within the region, at GRCh37 locus chr15:20,024,978. With the exception of the Rapid pipeline calling two separate A > T mutations, rather than a multiple nucleotide polymorphism (MNP) of AAA>TAT, all three pipelines report the same variants within this region. Notably in Figure 9, the hybrid and CPU pipelines show called variants with the exception the variant at site chr15:20,024,978 being 'grayed out', due to low quality flags being triggered in the GATK Variant Filtration quality control step (GATK (Best Practices), 2020). These low-quality variants are reported in the VCF and are discordant from the NIST HC variants, the low-quality flag in the 'FILTER' column allows for an analyst or automated filter set to easily pass over these variants. Despite having the highest raw sensitivity, specificity, PPV, and NPV, the Rapid GPU pipeline has no such variant filtering step, meaning any doubtful variants must be scrutinized more by genomic analyst. Given the superior speed of the GPU pipeline, it is more than reasonable to add an additional variant QC step to the end of the pipeline to reduce or flag low confidence variant calls in the resulting VCF.

Figure 9: An IGV screenshot of a low complexity genomic region showing the High Confidence callset and all three pipeline outputs. The clinical CPU and Hybrid pipelines show grayed out, 'filtered' variants that are called, but flagged as untrustworthy.



Clinical Efficacy

All three pipelines display the sensitivity, specificity, PPV and NPV that is required of a clinical pipeline. The Rapid GPU pipeline shows great promise for clinical work, showing the ability to decrease bioinformatic workflow turnaround times. This is especially useful in cases of neo-natal crises, or other emergent situations where an early detection/diagnosis of diseases could prevent irreversible damage (Kingsmore, 2012). In these cases, whole genome sequencing has become a first line of defense for clinicians unable to diagnose disease by conventional diagnostic methods (Bodian, 2014). The largest draw back for these cases is the turnaround time for genome sequencing and reporting – which can take upwards of a month. Implementing the Rapid GPU pipeline in these scenarios offers a massive speed up and could save lives in the process.

One of the notable drawbacks of the Rapid GPU pipeline as outlined in this study is the lack of built in quality control and quality assurance steps. While this is true, the data outputs from the Rapid pipeline are in standard bioinformatic formats can easily be picked up by CPU based quality control methods. CPU quality control steps can be applied to Parabricks GPU outputs, as demonstrated within the hybrid pipeline. In the cases of neonatal crisis, genomic analysis of variants could even begin immediately upon the completion of the Rapid GPU pipeline. Slower, CPU quality control steps could be performed in parallel to the initial clinical analysis. Once quality control steps complete, the results can be assessed accordingly. Combinations of the CPU and GPU methods could cut hours and even days from results generation for genomic workflows.

CONCLUSIONS

GPU based workflows are rapidly transforming the landscape of many HPC processes, especially in fields dependent on getting results quickly. Genomics and bioinformatics present a perfect application for using GPUs to speed up computational workflows due to their 'ridiculously parallel' nature. This study shows that Parabricks and its suite of genomics tools provide massive speed boost to current state of the art clinical workflows, cutting the time from job submission to results by four to five-fold. Some additional work needs to be done to Parabricks add quality control steps into the germline workflow. The 'Hybrid Pipeline' discussed within this study presents a reasonable method to both reduce turnaround time and makes use of quality control features present within the CPU based pipeline by adding CPU tools such as GATK's Variant filtration or VQSR (GATK (Best Practices), 2020).

Parabricks' speed and throughput increase for WGS processing also may provide a roundabout solution to the massive data volume produced by WGS pipelines. As Parabricks drastically reduces the compute time for creating and recreating genomic data, output data could be deleted after analysis and only the smaller, raw sequencing data files kept long term. This method would result in a drastic reduction of overall clinical data burden on clinical labs. By only keeping raw data and rapidly recreating results on request, clinical labs may be able to lower prices on WGS clinical tests and provide greater accessibility of WGS testing to patients.

In the context of clinical care, the speed of Parabricks GPU bioinformatic tools has been shown to reduce overall WGS bioinformatic pipeline processing time by days. Coupled with innovative analysis methods, such as starting variant analysis as soon as variant call files are ready, the processes Parabricks facilitates could cut up to a week from overall TAT. While seemingly minimal, the reduced turnaround time can save lives by diagnosing diseases and starting treatments or clinical trials sooner or assist clinicians and families in the decision to start palliative care sooner. The reduction in TAT can reduce the overall cost of treatment for patients by providing them a personalized care plan which can shorten inpatient hospital stays. Finally, quicker results can reduce stress for families and clinicians alike by providing the most comprehensive test possible in a shorter time frame.

Parabricks exhibits clinical efficacy showing high sensitivity, specificity, PPV and NPV that is ready for use right out of the box. While minimal quality control is included in its standard workflow, the Parabricks suite of tools provides configurable modules that can be included in combined CPU and GPU pipelines. Used wisely, Parabricks and GPUs can bring whole genome sequencing closer to a routine clinical test, rather than a last-ditch effort.

Future Studies

Future work surrounding the use of Parabricks and GPUs should be centered around applying additional quality control mechanisms to Parabricks Rapid GPU pipeline. One such direction would be the application of variant filter fields such as

GATK's VQSR, or GATK's Filter variants (GATK (Best Practices), 2020). Resulting Rapid GPU filtered VCFs then can be more meaningfully compared to one another compared against the filtered VCFs from the Clinical CPU pipeline. The resulting comparison would give additional weight to the clinical efficacy of the GPU pipeline. Another useful quality control feature that could be added to the GPU pipeline are DepthOfCoverage and Bam Statistics by BamTools (Barnett, 2011). These additional comparisons and quality control steps would add the necessary quality control to go live with WGS bioinformatic analysis by Parabricks GPU pipelines.

BIBLIOGRAPHY

Adaptive Computing. (2013, September). *TORQUE Resource Manager.* Retrieved from Adaptive Computing: http://docs.adaptivecomputing.com/torque/4-2-6/torqueAdminGuide-4.2.6.pdf

Barnett, D. W. (2011). BamTools: a C++ API and toolkit for analyzing and managing BAM files. *Bioinformatics*, 1691-1692.

Bodian, D. L. (2014). Diagnosis of an imprinted-gene syndrome by a novel bioinformatics analysis of whole-genome sequences from a family trio. *Molecular genetics & genomic medicine*, 2(6), 530-538.

CAP. (2020). College of American Pathologists: Clinical data. *Molecular Checklist*. United States: CAP.

Church, D. M. (2011). Modernizing reference genome assemblies. *PLoS Biol*, e1001091.

Cirulli, E. T. (2010). Uncovering the roles of rare variants in common disease through whole-genome sequencing. *Nature Reviews Genetics*, 415-425.

Clark, M. M. (2019). Diagnosis of genetic diseases in seriously ill children by rapid whole-genome sequencing and automated phenotyping and interpretation. *Science translational medicine*.

Danecek, P. A. (2011). The variant call format and VCFtools. *Bioinformatics*, 2156-2158.

DROAZEN. (2019, April 23). *broadinstitute / gatk*. Retrieved from GITHUB: https://github.com/broadinstitute/gatk/releases/tag/4.1.2.0

GATK (Best Practices). (2020). *(How to) Map reads to a reference with alternate contigs like GRCH38*. Retrieved from broadinstitute: https://gatk.broadinstitute.org/hc/en-us/articles/360037498992--How-to-Map-reads-to-a-reference-with-alternate-contigs-like-GRCH38

Hood, L. (2003). Systems biology: integrating technology, biology, and computation. *Mechanisms of ageing and development,*, 9-16.

Illumina. (2019, February). *bcl2fastq2 Conversion Software v2.20.* Retrieved from Illumina.com: https://support.illumina.com/content/dam/illumina-support/documents/documentation/software_documentation/bcl2fastq/bcl2fastq2-v2-20-software-guide-15051736-03.pdf

Illumina. (2020). *TruSeq DNA Nano*. Retrieved from Illumina: https://www.illumina.com/products/by-type/sequencing-kits/library-prep-kits/truseq-nano-dna.html

Kingsmore, S. (2012). Comprehensive carrier screening and molecular diagnostic testing for recessive childhood diseases. *PLoS currents*, 4.

Lander, E. S. (2001). Initial sequencing and analysis of the human genome. *Nature*.

Li, A. H. (2011). A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, 2987-93.

Lionel, A. C. (2018). Improved diagnostic yield compared with targeted gene sequencing panels suggests a role for whole-genome sequencing as a first-tier genetic test. *Genetics in Medicine*, 435-443.

Loeber, J. G. (2013). Newborn screening programmes in Europe; arguments and efforts regarding harmonization. Part 1–From blood spot to screening result." Journal of Inherited Metabolic Disease: . *Official Journal of the Society for the Study of Inborn Errors of Metabolism* , 603-611.

Manolio, T. A. (2017). Bedside back to bench: building bridges between basic and clinical genomic research. . *Cell*, 6-12.

Markov, I. L. (2014). Limits on fundamental limits to computation. *Nature*, 147-154.

Meienberg, J. B. (2016). Clinical sequencing: is WGS the better WES? *Human genetics*, 359-362.

Miller, N. A. (2015). A 26-hour system of highly sensitive whole genome sequencing for emergency management of genetic diseases. *Genome medicine*, 1-16.

Moore, G. E. (1998). Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 82-85.

Muir, P. L. (2016). The real cost of sequencing: scaling computation to keep pace with data generation. *Genome biology*, 1-9.

Muraskas, J., & Parsi, K. (2008). The cost of saving the tiniest lives: NICUs versus prevention. *AMA Journal of Ethics*, 655-658.

National Institute of Health. (n.d.). *Human Genome Project FAQ*. Retrieved from Genome.gov: https://www.genome.gov/human-genome-project/Completion-FAQ#:~:text=In%201990%2C%20Congress%20established%20funding,billion%20in%20FY%201991%20dollars.

National Institute of Standards and Technology. (2020, August 2). *Genome in a Bottle*. Retrieved from NIST: https://www.nist.gov/programs-projects/genome-bottle

Nickolls, J. a. (2010). The GPU computing era. *IEEE* , 56-69.

NVIDIA (CUDA). (2020, 6 23). *CUDA Toolkit Release Notes*. Retrieved from https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html

NVIDIA (Parabricks). (2020). *Clara Parabricks*. Retrieved from Clara Parabricks: https://developer.nvidia.com/clara-parabricks

Parabricks. (2020, 6 22). *clara-parabricks*. Retrieved from NVIDIA Clara Parabricks: https://developer.nvidia.com/clara-parabricks

Quinlan, A. R. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 841-842.

Rimmer, A. P. (2014). Integrating mapping-, assembly-and haplotype-based approaches for calling variants in clinical sequencing applications. *Nature genetics*, 912-918.

Rossen, J. W.-G. (2018). Practical issues in implementing whole-genome-sequencing in routine diagnostic microbiology. *Clinical microbiology and infection*, 355-360.

Saunders, C. J. (2012). Rapid whole-genome sequencing for genetic disease diagnosis in neonatal intensive care units. *Science translational medicine*, 154.

Schwarze, K. B. (2018). Are whole-exome and whole-genome sequencing approaches cost-effective? A systematic review of the literature. *Genetics in Medicine*, 1122-1130.

Sun, X.-H. a. (2010). Reevaluating Amdahl's law in the multicore era. *Journal of Parallel and distributed Computing*, 183-188.

Thiffault, I. F. (2019). Clinical genome sequencing in an unbiased pediatric cohort. *Genetics in Medicine*, 303-310.

Van der Auwera, G. A.-M. (2013). From FastQ data to high-confidence variant calls: the genome analysis toolkit best practices pipeline. *Current protocols in bioinformatics*.

Waldrop, M. M. (2016). The chips are down for Moore's law. *Nature News*, 144.

Willig, L. K. (2015). Whole-genome sequencing for identification of Mendelian disorders in critically ill infants: a retrospective analysis of diagnostic and clinical findings. *The Lancet: Respiratory Medicine*, 377-387.

Zhao, M. W. (2013). Computational tools for copy number variation (CNV) detection using next-generation sequencing data: features and perspective. *BMC bioinformatics*, 14.

Zook, J. M. (2014). Nature biotechnology. *Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls*, 246-251.

Zook, J. M. (2019). An open resource for accurately benchmarking small variant and reference calls. *Nature biotechnology. 37(5)*, 561-566.