Microsoft Genomics

Accelerate precision medicine with Microsoft Genomics

The cost of sequencing a human genome has dropped dramatically, from millions of dollars a decade ago, to only a thousand dollars. This has enabled research programs to sequence hundreds of thousands of people, and clinical programs to sequence patients' genomes as a standard part of their treatment process. This dramatic expansion of genome sequencing also demands a lot of data storage and computing power, since each genome is about 60 Gb of compressed data, and normally requires about a thousand CPU-hours to process.

The Microsoft Azure cloud is ideally suited to satisfy this demand, with reliable, secure, global data storage and computation services. Microsoft Genomics service on Microsoft Azure provides an easy-to-use web service for analyzing genomes that is several times faster than the standard genomics pipeline. This service follows the best practices for concordance and accuracy established by the Broad Institute of MIT and Harvard, the de facto standard for genomic analysis. The speed, accuracy, and simplicity of the Microsoft Genomics service enables a wide range of applications in cancer, rare diseases, population health, and precision medicine.

1. Introduction

The DNA in a normal human cell consists of 23 pairs of chromosomes, adding up to 6.4 billion base pairs (along with a few other bits, like mitochondrial DNA). Each base pair is represented by the letter A, C, T, or G identifying the specific nucleotide at each location. Since the paired chromosomes are very similar, they are typically lined up together, numbered from 1 (largest) through 22, then the X & Y sex-specific chromosomes, into a sequence of 3.2 billion locations. Since human DNA varies relatively little among individuals – typically about 1 in 1000 locations – we usually compare a given person's DNA to a reference genome, a composite genome based on the results of the first Human Genome Project.

The process of sequencing a genome starts with a biological sample, such as blood or saliva, and ends up with a report of the differences from the reference genome and what they mean. This is commonly divided into three stages:

1) Primary analysis, which analyzes a sample biochemically and produces raw data

2) Secondary analysis, which takes the raw data, aligns it to the reference, and identifies variants (differences) from the reference

3) Tertiary analysis, which analyzes the variants and adds annotations to help interpret their biological or clinical implications

The primary analysis stage is done in the laboratory using specialized sequencing instruments from companies such as Illumina and Thermo-Fisher. The genome sequencer replicates and fragments the DNA and then reads the base pairs, in a massively parallel process combining biochemistry, optics, electronics, and image processing. Identifying the base pairs in a sequence of DNA (base calling) is hard and errors do happen, so in addition to the A, C, T, or G base called for each location, the sequencer also produces a 'quality score' to record how confident the sequencer is in the base call. Sequencing an entire human genome typically produces about a billion roughly 100-character strings ("reads") of A, C, T, and G, covering the genome with an average of 30 copies for redundancy. Along with other metadata such as quality scores, this produces about 60 GB of compressed raw data ready for secondary analysis.

The first step in secondary analysis is to align each read to the reference genome, finding the closest match among 3 billion possible locations, allowing for errors in the sequencing process, differences between the sample and the reference genome, and the presence of many similar regions across the span of the genome. The next step is variant calling, which looks at the differences between the reference and the reads aligned to each location, and decides whether they are errors in sequencing, or true variants in the sample DNA. These variants can be simple single-nucleotide variants (SNVs), which are substitutions of one A, C, T, or G for another, or more complex insertions, deletions or rearrangements.

Tertiary analysis is more complex and varied. There is a wide variety of tools and databases in use for this stage. Depending on the purpose of the analysis, these might add information about evolutionary conservation, protein structure, drug response, disease risk, genomic interactions, etc. The choice of databases and tools depends greatly on the purpose of the analysis, and the overall methodology of the clinician or researcher.

Regardless of the kind of tertiary analysis required, secondary analysis is always necessary. Both the alignment and variant calling steps in secondary analysis are very computationally demanding, because they need to process a large amount of data and perform complex calculations. The standard tools for secondary analysis are the Burrows-Wheeler Aligner (BWA) for alignment, and the Genome Analysis Toolkit (GATK) for variant calling, developed by researchers at the Broad Institute of MIT & Harvard and the Sanger Institute in the UK. The normal versions of these tools take over a day to process a 30x whole genome sample on a 16-

core server, starting with the raw read data from the sequencer, and producing a file of aligned reads and a file of variant calls.

2. Microsoft Genomics

The growing scale, complexity, and security requirements of genomics make it an ideal candidate for moving to the Microsoft Azure cloud. Microsoft Azure has datacenters around the world, with the storage and compute power to meet the demands of storing and analyzing the hundreds of thousands of genomes that will be sequenced in the coming years. Microsoft Azure is accredited to comply with the major global security and privacy standards, such as ISO 27001, and has the security and provenance standards that enable HIPAA-compliant operation when handling personal health information.

As part of making Microsoft Azure the best cloud for genomics, Microsoft Genomics has developed an optimized secondary analysis service that can process a 30x genome in only a few hours, instead of a day or more. The Microsoft Genomics Service includes a high-performance engine that is optimized to read large files of genomic data, process them efficiently across many cores, sort and filter the results, and write them back out. This engine orchestrates the operation of the BWA aligner and the GATK HaplotypeCaller variant caller for maximum throughput. It also incorporates several other simpler components that are part of standard genomics pipelines, such as duplicate marking, base quality score recalibration, and indexing. This engine can process a single genomic sample, from raw reads to aligned reads and variant calls, in a few hours on a single multi-core server.

The Microsoft Genomics service controller manages the processing of batches of genomes distributed across pools of machines in the cloud. It maintains a queue of incoming requests, distributes them to servers running the genomics engine, monitors their performance and progress, and evaluates the results. It ensures that the service runs reliably and securely at scale, behind a secure web service API. Clients don't need to deal with the complexity of managing and updating their hardware and software, and can rely on fast, efficient execution of an accurate best-practices genomics pipeline. The results can be easily connected to tertiary analysis and machine learning services, such as Microsoft R Server on Azure.

2.1. Client Architecture

The Microsoft Genomics client (msgen) is a Python front-end to the web service. It can be installed like a standard Python package, on Windows or Linux using the Python pip package manager ("pip install msgen"). For each genome sample that you want to process, you create a configuration file containing all the parameters

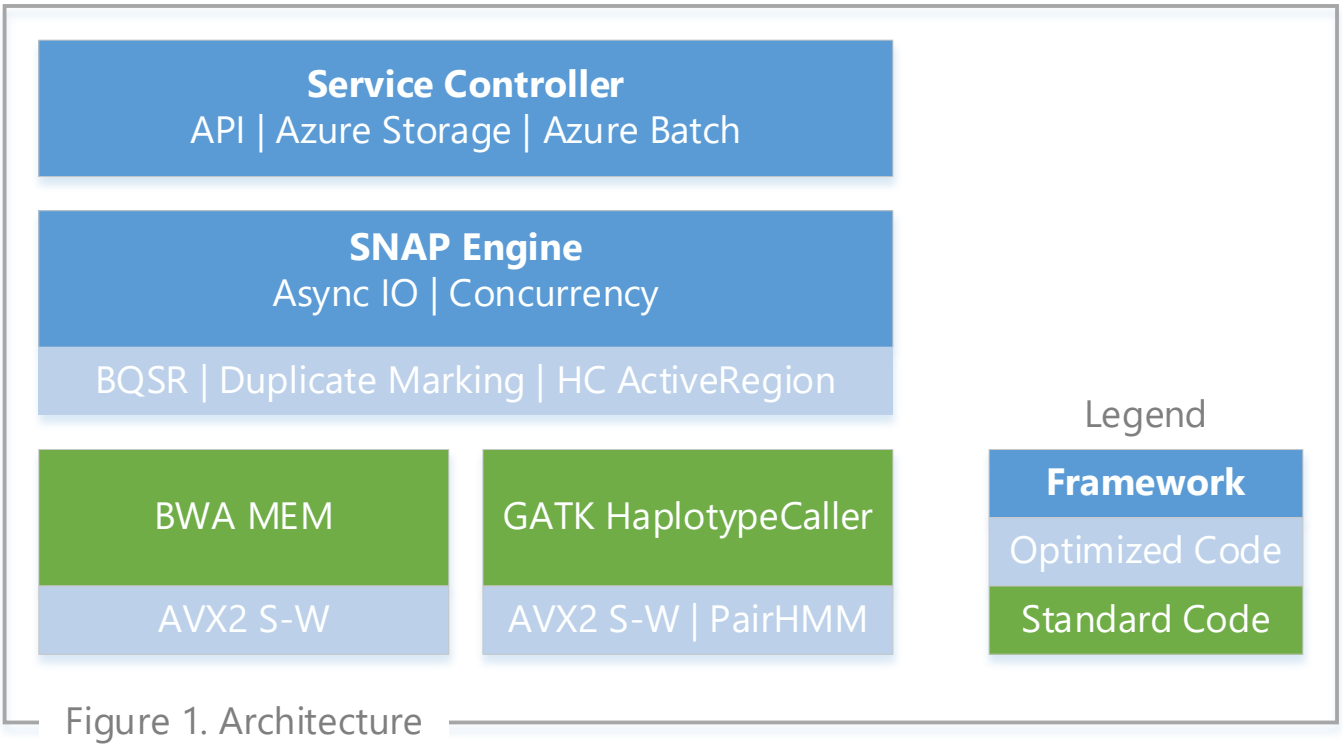

Figure 1. Architecture

for downloading the data, running the Microsoft Genomics pipeline, and uploading the results:

- Your subscription key to Microsoft Genomics
- The process to run and its parameters
- Path information and storage account keys for the input files in either paired FASTQ, paired compressed FASTQ, or BAM format, in Azure Storage
- Path information and storage account key for the location to place the output files in Azure Storage

You can then invoke the msgen client to initiate processing, and monitor progress until the job is complete. The final aligned reads in BAM format, and variant calls in VCF.GZ format will be placed in your designated output container in Azure Storage. The client can easily be incorporated into existing workflows.

2.1. Service Architecture

The Microsoft Genomics service is responsible for processing the genomic data in Azure. The overall system architecture consists of several layers, as shown in Figure 1:

1) The service control layer, for receiving API requests, scheduling the work queue, and managing execution across pools of machines in Azure Batch.
2) The SNAP execution engine, for orchestrating the IO and computation of a single sample on a single machine from beginning to end.
3) Optimized versions of GATK pipeline pre- and post-processing steps, such as

duplicate marking, base quality score recalibration, indexing, and BAM compression.
4) Standard BWA-MEM and HaplotypeCaller algorithms for alignment and variant calling, with minimal modifications to enable optimizations while retaining compatibility.
5) Optimized AVX2 code for compute-intensive algorithms such as Smith-Waterman sequence alignment, and Pair Hidden Markov Model for haplotype evaluation.

2.1.1. Service Controller

The service controller is a distributed C# web application with a back-end service executable. The front-end accepts client requests from Azure API Management, and places an entry in a work queue. An Azure Web Job application then schedules & monitors an Azure Batch task for each queue item. When Azure Batch executes the task, the service executable downloads the reference data and input files, runs the SNAP engine for alignment & variant calling, streams the resulting files back to Azure Storage as they are being written, and reports completion. The overall application follows Azure best practices for security, compliance, auditing, and monitoring (for example, all client and application secrets are kept in Azure Key Vault for protection).

2.1.2. SNAP Engine

The SNAP engine is based on the high-performance SNAP short read aligner developed by Microsoft Research in collaboration with the UC Berkeley AMPLab. This version does not use the SNAP alignment algorithm, but has replaced it instead the more widely used BWA MEM aligner. It has a high-performance asynchronous input/output subsystem for efficiently handling large volumes of genomic data. It also includes an efficient system for scheduling compute-intensive work across multiple cores and gathering the results. The overall data flow through the framework is designed to minimize unnecessary disk traffic, reading and writing the data in only two passes rather than the half-dozen or more required by the standard BWA/GATK pipeline.

The first pass aligns the reads in large batches concurrently across all cores, and does any preprocessing that does not depend on ordering, such as gathering statistics for base quality score recalibration. The reads are then written to an intermediate file in large sorted batches. The second phase merges all the batches into a single fully-sorted stream of reads, and finishes the pre- and post-processing steps – applying BQSR statistics, marking duplicates, building the BAM index, and compressing the BAM file. Concurrently with writing the BAM file, this phase also orchestrates several GATK processes to do variant calling, identifying the active regions around potential variants, and piping only those reads directly into HaplotypeCaller for efficiency. Since sorting is IO-intensive and variant calling is CPU-intensive, this efficiently balances overall use of machine resources.

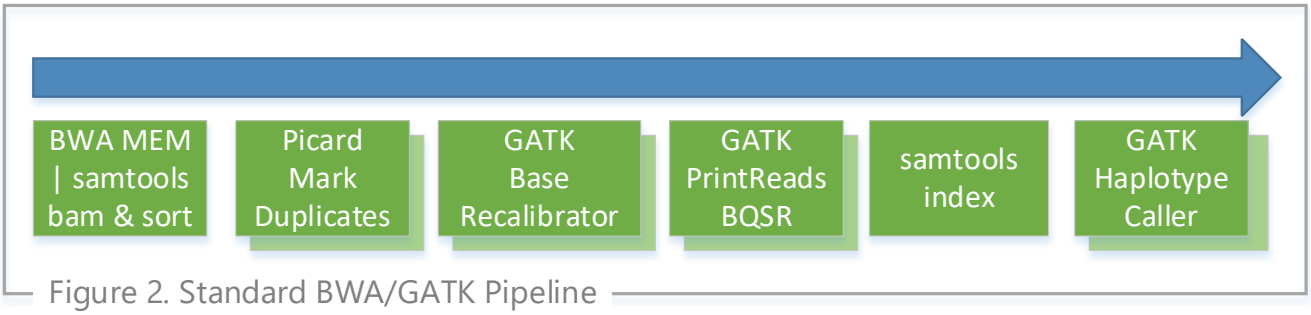

Figure 2. Standard BWA/GATK Pipeline

2.1.3. BWA-MEM & GATK HaplotypeCaller

Microsoft Genomics started with the open-source version of BWA, and a licensed version of GATK from the Broad Institute, and then optimized and accelerated them to run on the Microsoft Azure cloud. The bulk of the BWA-MEM and GATK HaplotypeCaller code is unchanged to preserve compatibility with standard pipelines. There are only a few changes to meld them into the overall pipeline – BWA-MEM has been made into a library, and HaplotypeCaller has been modified to accept pre-calculated active regions piped into standard input. Also, the duplicate marking and base quality score recalibration (BQSR) algorithms from Picard and GATK have been translated into C++ and are applied as the data is streamed through the engine to minimize unnecessary disk I/O.

The bulk of the time in these programs is spent in a couple of low-level compute kernels, Smith-Waterman and PairHMM. These have been highly optimized using Intel AVX2 vector instructions to run significantly faster than the standard versions. The initial release is planned to support GATK 3.5, with support for other versions to be added over time based on customer demand. The pipeline has an option to produce a gVCF file, which can be merged across multiple samples for joint genotyping.

2.2 Performance

The overall architecture of the Microsoft Genomics service can scale to processing hundreds or thousands of genomes in parallel, by elastically allocating resources from the Microsoft Azure cloud. Each genome is run separately on a single high-capacity virtual machine, to maximize throughput and minimize communication & storage overhead within each run. The pre- and post-processing steps are run in parallel as much as possible, to reduce end-to-end processing time.

The standard GATK Best Practices pipeline normally runs each step sequentially. Each step reads the files produced by the preceding step, and so must wait for it to complete. There might be parallelism within a single step, where the work can be divided into distinct regions of the genome (e.g. running MarkDuplicates across each chromosome separately), as shown in Figure 2.

The Microsoft Genomics pipeline looks quite different. It only makes two passes over the data,

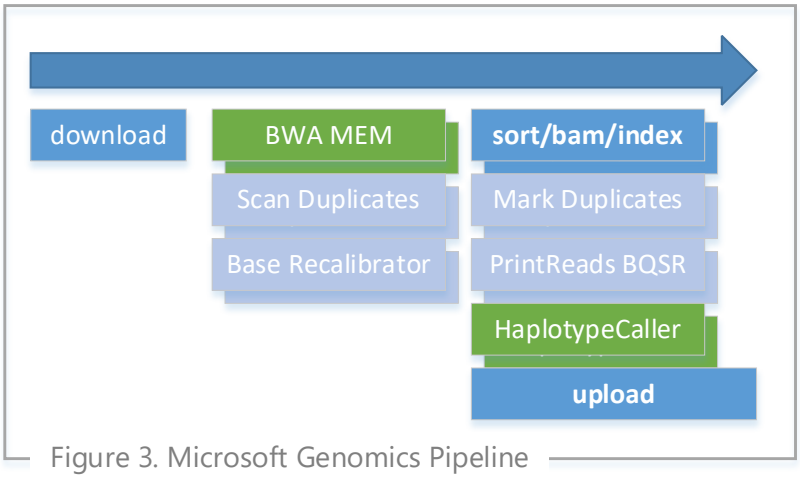

Figure 3. Microsoft Genomics Pipeline

combining many different steps of processing in each pass, and also processing many regions in parallel, for maximum efficiency, as shown in Figure 3.

2.3. Concordance

The Microsoft Genomics service produces results that are highly concordant with the standard BWA/GATK pipeline. Figure 4 compares the results of the Microsoft Genomics service with the output of the Broad best practices BWA/GATK pipeline, broken down in two ways:

- Aligning + preprocessing + variant calling – Entire pipeline of alignment, duplicate marking, quality score recalibration, and variant calling, starting with the same FASTQ files, and comparing the resulting VCF files
- Variant calling – Just the final variant calling step, comparing the VCF files from the Broad version of GATK HaplotypeCaller with the Microsoft Genomics accelerated version, on the same aligned, preprocessed BAM file

It shows the F1 score for matching variants, which is a composite score of precision and recall, across several whole-genome runs of different samples. Figure 4 shows both the results across the whole genome, as well as the calls restricted to the NIST Genome in a Bottle "high-confidence" regions, where the results of variant calling are significantly more reliable and reproducible. The differences are comparable to the variation between runs of GATK due to random number generation, multiple threads, etc.

Figure 5 shows a comparison of the Microsoft Genomics and Broad BWA/GATK pipelines with the NIST Genome in a Bottle truth set, comparing only variants within the high-confidence regions. The results are essentially identical, with a slight improvement in the Microsoft Genomics results.

The Microsoft Genomics platforms also produces very consistent results from run to run. Figure 6 shows the variability of the number of variants produced over the course of 20 runs on several different samples from the 1000 Genomes and Illumina Platinum Genomes datasets.

Figure 7 shows similar run-to-run comparisons for the percentage of matching variants (i.e. with same CHROM, POS, REF, ALT), and the percentage of matching variants with identical genotypes.

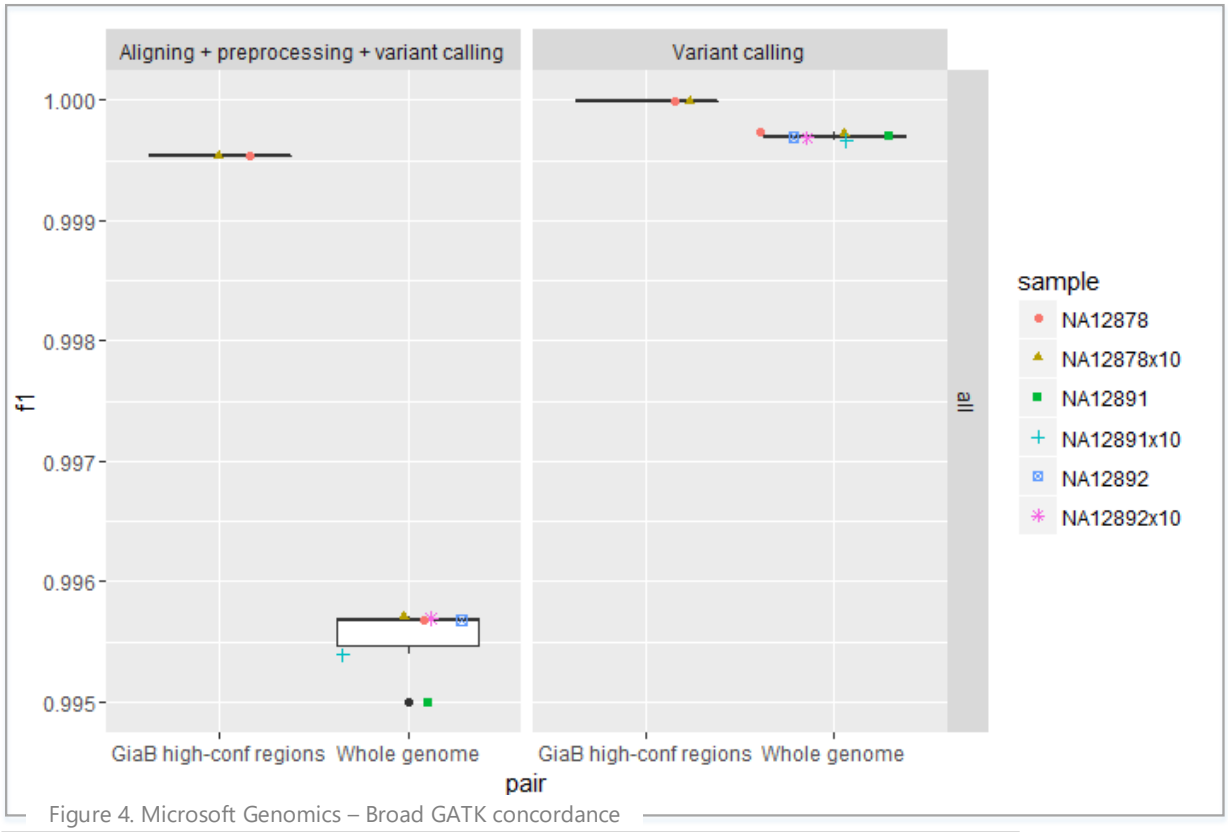

Figure 4. Microsoft Genomics – Broad GATK concordance

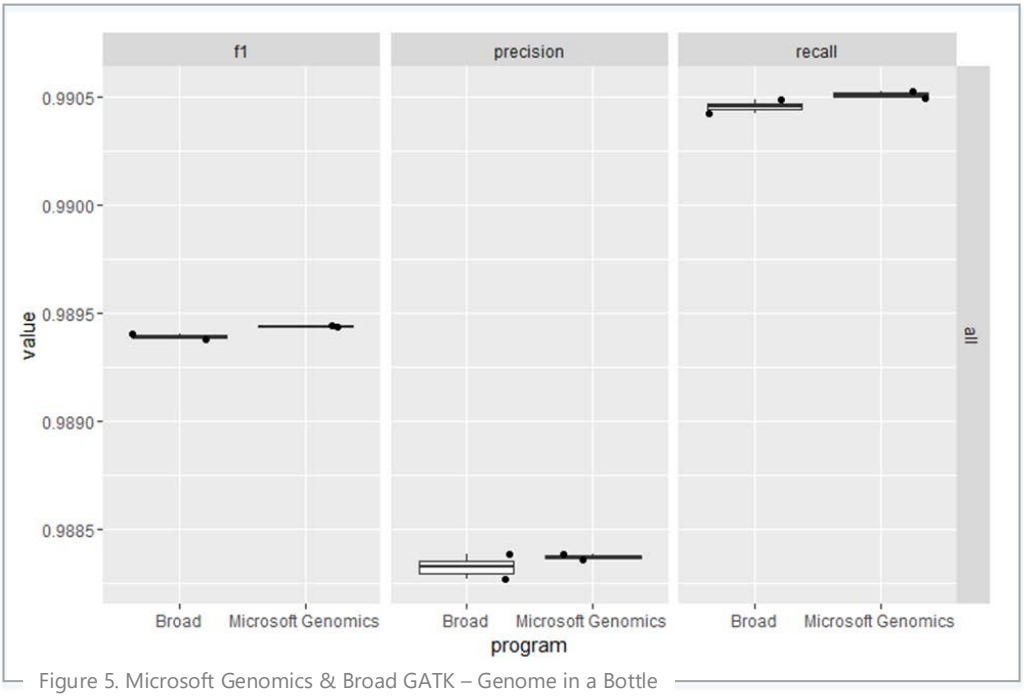

Figure 5. Microsoft Genomics & Broad GATK – Genome in a Bottle

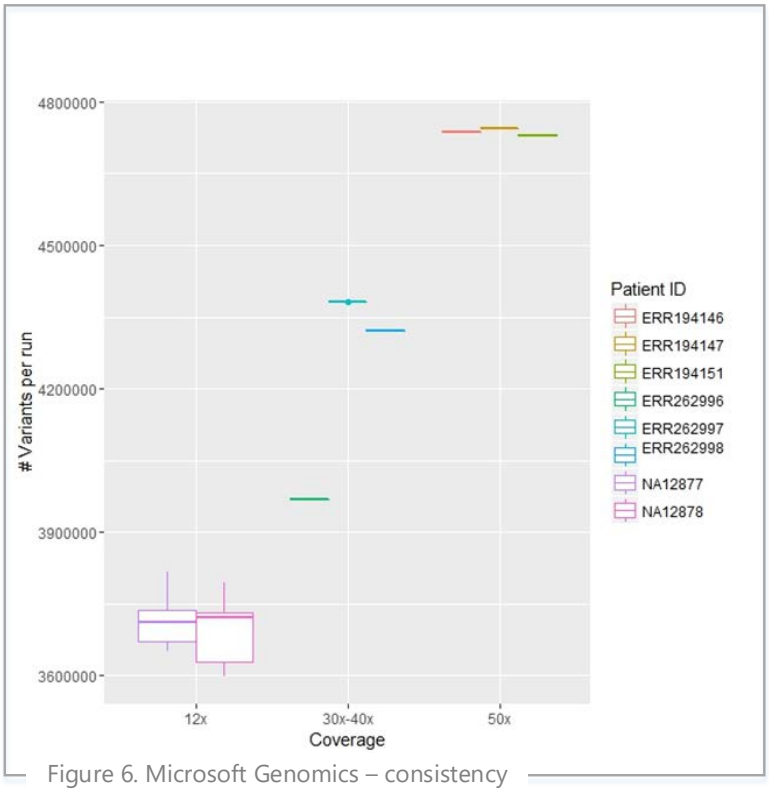

Figure 6. Microsoft Genomics – consistency

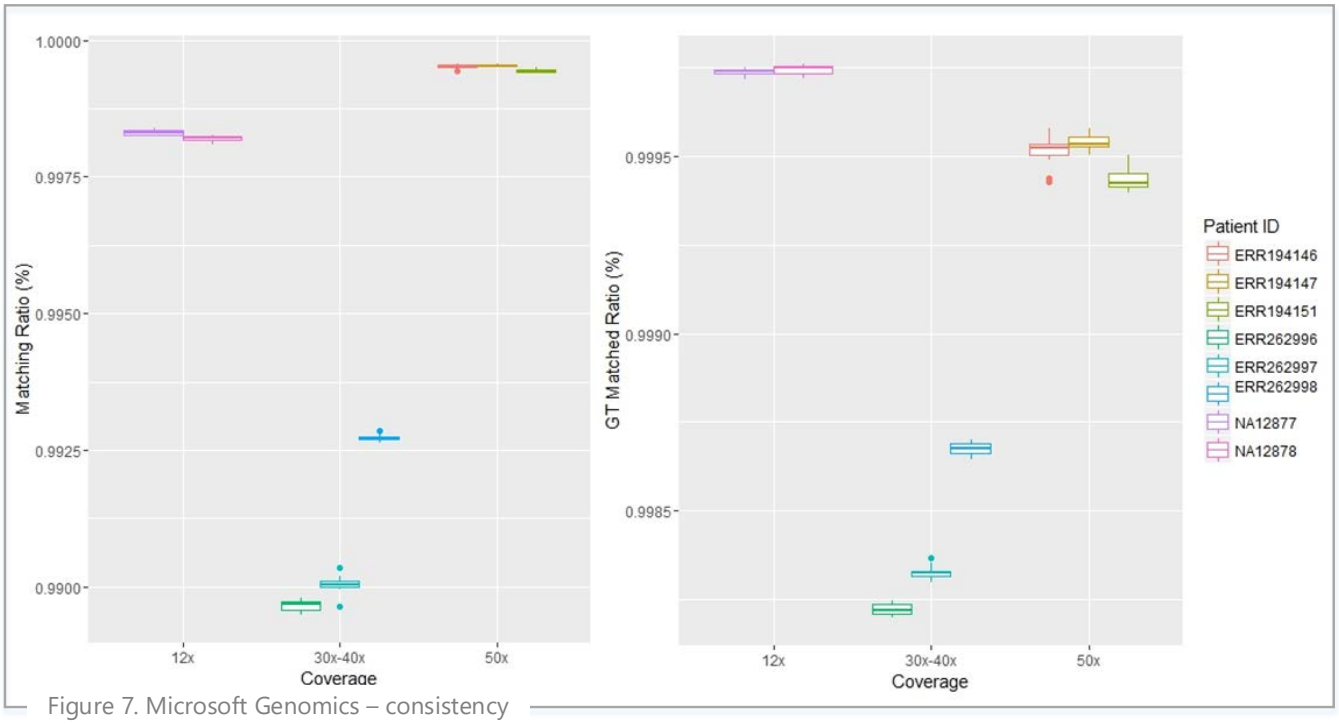

Figure 7. Microsoft Genomics – consistency

Table 1 shows the concordance of just the alignment stage of the pipeline, comparing the aligned BAM from Microsoft Genomics with that produced by BWA MEM. 97.02% of the reads have the same alignment. Of the remaining 2.98% of reads, 2.46% have MAPQ<20 and are filtered out by standard variant calling pipelines, and 0.36% have the same alignment after adjusting for soft clipping. This leaves only 0.16% differing alignments with MAPQ>=20, which has little effect on concordance, as shown by the variant calling concordance results.

Match Criteria	Matching reads
Alignment match	97.02%
… OR MAPQ < 20	99.48%
… OR adjusted softclip	99.84%

Table 1. Alignment concordance