NGSomics (Next Generation Sequencing Genomics): A Mobile-optimized, Cloud-based NGS Pipeline that Utilizes Docker Containers for Parallelization and Automation

Introduction

Question:

Can the entire process of DNA analysis, from Next-Generation Sequencing to Variant calling to bedside applications, be simplified and transformed into a simple cloud-based, mobile platform, which not only chains multiple tools together in one automatic pipeline, but also display meaningful and accurate results easily, to enthrone the next era of personalized medicine?

Background Research:

[Insert diagram of workflow/importance]

What is DNA Sequencing?
DNA sequencing is the process by which the order of nucleotides of a DNA molecule are determined – where the 4 base pairs (A, G, C, T) are in a strand of DNA.[1,3] DNA sequencing is the basis on which genomic analysis is founded – it provides the foundation for how significant genetics-related factors are determined, including alignment and variant calling.[1,3]

Brief Historic Overview
[Insert timeline with basic descriptions]
Sanger Sequencing : 1977 (Frederick Sanger)
Polymerase Chain Reaction: 1983 (Kary Mullis)
Human Genome Project Starts: 1990
Next-Generation Sequencing : 2005

DNA Analysis Today
[Workflow image]

High Level Overview
The goal of any DNA analysis workflow is to detect significant variants from sequencing data. This often takes place after the DNA from the target organism is extracted and processed, a process in which the target DNA is fragmented and captured into beads, and with the help of cDNA (copy-DNA) and fluorescent dye, is transcribed into a large text files for variant calling and analysis.[2] These reads, or raw DNA base pair sequences extracted, are either classified as short, which contain 35-150 base pairs, or long reads, which contain 200-1000 basepairs).[4,7]

Many current DNA analysis pipelines specialize in one or the other, but NGSomics's workflow covers both of these options.
[Image = http://puu.sh/nDleS/150cb1ea41.png]

Detailed Process
FASTA, FASTQ (sequencing) → SAM/BAM (alignment) → VCF (variant calling)
[workflow image]
Sequencing: DNA assembly and sequencing is the process by which the base reads from the sequencing data are aligned and matched with the base pairs in the reference genome.[6] This process involves DNA mapping, a method which allows the shotgunned reads (isolated reads covering the same genomic structure - a key element of Next-Generation Sequencing) to be corresponded to the appropriate position on the reference genome.[2] This sequencing process takes the raw FASTA / FASTQ sequencing files (FASTQ for quality score) and a reference genome text file in order to perform the necessary alignment to create a SAM file (Sequence Alignment Map).[4]
[image of FASTQ file: https://puu.sh/nDlBc/18b5f42be2.png]
FASTQ files also contain a quality score, the Phred value / Q score = probability that the base is wrong.
[Q = -10log(e), so Q10 = 90% confidence (10% wrong), Q30 = 99.9% confident]

Alignment: After the DNA sequencing is finished, the SAM alignment file is compressed into a BAM file (Binary Alignment Map) and is sorted to create a more convenient format for variant calling programs to access.[5] This stage often involves the removal of any duplicated sequences.
[Insert IGVpicture of BAM file mapping]

Variant calling:
Variant discovery is the process by which statistically significant variants are found, and are commonly denoted with a few forms.[3,6] SNPs (Single Nucleotide Polymorphisms) are the substitution of one nucleotide base for another.[2] Indels are the deletions or additions of a nucleotide base.[2] Other less common ones include Inversion, a switch between two or more nucleotides, and Duplication, a copy of one or more nucleotides. Variants are found by comparing the alignment file to the sequenced genome, and variant calling programs identify areas in which a change is likely. This step converts the SAM alignment file into VCF files (Variant Call Format), which specify the location of a possible variant.
[Insert VCF file: https://puu.sh/nDlZn/87f5a0a05a.png]

Annotation and Evaluation:
Evaluation: To reliably identify variants is a daunting task, and lends itself to an inspection of the VCF file's accuracy:
True Positives (TP) are variants which are correctly identified, whereas **False Positives (FP)** are variants which were incorrectly identified. **True Negatives (TN)** are variants which are neither identified nor present in the organism, whereas **False Negatives (FN)** are variants which are present in the organism but not identified.

The combination of these two measurements to determine the variant caller's accuracy is the **False Discovery Rate**, a ratio of false positives to all variant calls made by the software. The FDR value interprets how many variants are likely to be false positives. **FP / (TP + FP)**
Specificity: True positive rate: ratio of correctly identified variants to total known variants in reference set. Higher sensitivity means it is more likely for a variant in the sample will be identified by the program. **TP / (TP + FN)**
Sensitivity: True negative rate: ratio of non-variant calls to the total number of positions in the reference set that are known to be homozygous with the reference sequence. Inversely related to the number of false positives. **TN / (TN + FP)**
[Insert sample GCAT test]
Annotation: After the accuracy of the variant calls are affirmed, meaning must be interpreted from the variants in the DNA sample – what are the effects of these variants in the organism? This step offers a variety of possible paths – the user can choose to identify patterns between variants for a particular illness, the phenotype most strongly affected by certain variants, and others. NGSomics creates its annotation capability by correlating the variants with possible drug treatments.
[Insert picture of drug dB, considering: https://puu.sh/nDmtH/5ab2480844.png and https://puu.sh/nDmuQ/8610f7b27f.png]

Why perform genetic-based analysis?
The increased interest in DNA analysis is largely due to the approaching reality of personalized medicine – where an individual's health problems can be easily and most efficiently addressed based on his/her DNA. However, DNA analysis has surpassed expectations; it contains applications in nearly every field of science, including (but not limited to):

- Identification of new/driver genes in certain genetically-based illnesses
- Increased precision in precision of location-based drugs
- Identification of evolutionary similarities between organisms
- Forensics and paternity tests
- **Disease correlations and personalized medicine**

Hypothesis:

While DNA analysis requires copious amounts of processing power and is inaccessible to ordinary people, ultimately the prospect of identifying illnesses based on one's own genetics in real time issues a new era of more targeted and otherwise personalized medicine; therefore, a more convenient and efficient software can be created to not only accomplish the tasks of current DNA analysis pipelines but also extend its functionality into the daily lives of many people, branching multiple aspects of engineering in order to serve as an applicable tool to professionals and commoners alike.

Goals & Objectives:

NGSomics – All your genes customized to fit in one jean pocket.
NGSomics is a complete genome analysis pipeline, which with both accuracy and speed, can be easily accessible and meaningful to the world of personalized medicine. NGSomics is unique with its cloud-based platform, specifically developed so that it is mobile-ready and virtually accessible to anyone with an internet connection and an electronic device. With the ever-developing world of genetic analysis, new solutions must be easy to improve and implementable on a large scale, and with Dockerized containers, NGSomics is not only very scalable, but also offers an automated pipeline, capable with running different genomic programs in parallel for a more accurate analysis. Genomic data is always convoluted with enormous text files that mean nothing to the ordinary person; however, with NGSomics's drug correlation and gene modeling capabilities, including functionality implemented to help display the results of a variant calling test, the results become very easy to visualize and make the tool easy to use and highly relevant.

Materials & Procedure

NGSomics is built on the open-sourced works of alignment, variant-calling, and annotation algorithms.

Alignment Tools:

BWA (Burrows-Wheeler Aligner)**: http://bio-bwa.sourceforge.net/**
BWA is used to map sequences to a reference genome, and it creates indices for the raw files to be aligned with the reference FASTA files.

```
$ bwa index -a bwtsw ref.fa
        #Indexes the reference FASTA file for processing raw reads
$ bwa aln ref.fa raw1.fq > reads1.sai
        #Finds SA coordinates of raw reads 1 (forward reads)
$ bwa aln ref.fa raw2.fq > reads2.sai
        #Finds SA coordinates of raw reads 2 (reverse reads)
$ bwa sampe ref.fa reads1.sai reads2.sai reads1.fq reads2.fq > aligned.sam
        #Aligns the raw reads with the reference file in order to output the "aligned.sam"
```

[image = https://puu.sh/nGLP7/4ef9f5c70c.png]

Variant Callers:

SAMtools: http://www.htslib.org/

Samtools is a genomic analysis tool widely used to perform a multitude of genomics functions on an alignment file. It requires an input SAM file and can call variants to output a BCF file (Binary Call Format). The Samtools package also contains BCFtools, which helps synthesize the results of SAMtools into VCF files. The SAMtools package was chosen due to its robust documentation and its popularity among bioinformaticians.

Sorting the alignment file (Needed for most other variant calling programs)

```
$ samtools view -b -S -o aligned.bam aligned.sam
        #Compresses the "aligned.sam" file into a more readable format "aligned.bam"
$ samtools sort -T aligned.sorted -o aligned.sorted.bam aligned.bam
        #Sorts the "aligned.bam" and outputs a sorted "aligned.sorted.bam" file
```

Variant calling

```
$ samtools mpileup -g -f reference.fasta aligned.sorted.bam > variants.bcf
        #Scans all aligned reads with the "reference.fasta" FASTA file to prepare the file
        for variant calling, outputting a binary call format file "variants.bcf" which displays
        genotype likelihoods
$ bcftools view variants.bcf > variants.vcf
        #Identifies specific variants from genotype likelihoods and outputs the final
        "variants.vcf" file
```

Varscan: https://dkoboldt.github.io/varscan/

Varscan is a variant caller program which detects SNPs, Indels, and other types of variants from the sorted alignment file.

```
$ samtools mpileup -f reference.fasta aligned.sorted.bam | java -jar varscan.jar
mpileup2snp > variants2.vcf
        #Uses the output of SAMtool's mpileup function in order to call, in this case,
        SNPs and outputs the "variants2.vcf" file.
```

Freebayes: https://github.com/ekg/freebayes

Freebayes is a Bayesian-based (inferential probability for variant call) variant caller program, which we chose because it is based on a haplotype caller. A haplotype caller focuses on related polymorphisms within a genomic structure as opposed to relying more on the alignment of the sequence.

```
$ freebayes -f reference.fasta aligned.sorted.bam > variants3.vcf
        # Uses the input sorted BAM file "aligned.sorted.bam" to scan to the reference
        "reference.fasta" file in order to output the VCF file "variants3.vcf""
```

Variant Annotation:

VCFtools: https://vcftools.github.io/

VCFtools provides applicable commands to analyze complex VCF files, including limiting the results to just indels or SNPs. This package was extremely relevant due to its ability to identify similarities and compare over 2 VCF files at a time, thus serving as an endpoint for the final VCF files for the NGSomics's pipeline.

```
$ vcftools --vcf variants.vcf --vcf variants2.vcf --gzdiff variants3.vcf --diff-site
```

#Uses the 3 VCF files from the variant-caller programs in order to identify differences between these files, outputting a final “out.diff.site_in_files” which displays the area and contents of the variant.

[Insert final output picture here = https://puu.sh/nEmVK/09a1d12cc3.png]

Iobio: http://iobio.io/

Iobio allows for easy visualization of genomic data, from VCFs to BAM files. NGSomics utilized its vcf iobio bundle in order to help visualize the final output variant-calling data from the pipeline.

[Insert picture of iobio src = https://puu.sh/nGGCH/724d1b4c10.png]

Cloud optimization:

Every component of NGSomics runs on a server in the cloud, which allows NGSomics to avoid being restricted by processing limitations, lack of memory, and insufficient storage. It also enables any user to access a suite of NGS programs from any computer with an internet connection. Finally, unlike a physical computer, it is easily expandable since resources can be increased with ease.

Docker:

Docker is a core component of the pipeline. It allows for containerization, since each individual program is run as its own isolated instance and thus can be replicated, scaled, and allocated resources easily. Since each individual component of the program interacts with others, NGSomics utilizes Docker’s volumes feature to enable shared files and data transfer.

Node:

NGSomics uses the Digital Ocean server to run Node.js, which maps a Docker container port to a public port. This allows users to access an intuitive front-end interface which sends input files to our Node.js server and converts returned output data into a readable format.

DigitalOcean: processing and accessibility from any terminal/browser

DigitalOcean provides hosting services, speedy machines, and can handle a large number of concurrent connections. In addition, it has servers around the world which enables users from anywhere to access and use the NGSomics pipeline. DigitalOcean also enables easy re-allocation of resources for an affordable price.

Design:

Server:

http://puu.sh/nHatF/5c79449c94.png

A Node.js server runs constantly, automatically updating when changes are detected. This server provides routes for a client (browser) to send data to and receive data from. In the case

of NGSomics, the client uploads reference files to the pipeline, and receives the program's output files as a result. The Node.js server automatically moves input files into a new directory organized by date of upload, and starts the pipeline based on the directory's location.

Docker workflow (shell script + nextflow):

http://puu.sh/nGTSu/c42b3a1350.png

The Docker workflow represents the core element of the pipeline; it not only allows for an easy way to scale and increase the functionality of the attached programs, but also allows for an easy conversion between the cloud and user-uploaded files. Since it needs to work for multiple concurrent users, it takes in directory name as input from the node script. It first prepares the input files by unzipping them and moving them to the appropriate directory. It then runs BWA (Burrows-Wheeler Aligner) along with various tools from the Samtools package. It finally outputs the data into a readable format for visualization through Iobio and other tools.

Drug Correlation

http://puu.sh/nHbsc/d2133fa657.png

Limitations of Current Designs

A major limitation of the Sanger sequencing technique, which is characterized by its limit to 100-1000 length long reads, is that it analyzes segments of base pairs individually. Next-Generation Sequencing techniques overcome this limitation by using "shotgun sequencing" to break DNA into numerous segments and analyze them all at once. NGS software also utilizes confidence tests to map the base pairs correctly with the reference genome.

Current designs also require a heavy amount of processing power. On the other hand NGSomics utilizes a 64GB Ram (20 CPUs), 640 GB SSD disk, 9TB Transfer Bandwidth in order to smoothly handle several concurrently running programs and users. In addition, current designs are individualized and isolated, and thus cannot work together in parallel with other machines. Finally, current techniques require a large amount of background knowledge and familiarity with complex command line tools.

https://puu.sh/nGLP7/4ef9f5c70c.png

How ours addresses these limitations

- **Speed & Processing**: NGSomics offloads all processing tasks to a processor in the cloud. In test conditions, NGSomics used 64GB Ram with 20 CPU cores to speed up test results.
- **Mobile**: NGSomics' front-end interface is mobile-friendly, and can be used across browsers and devices, including a smartphone.
- **Cloud**: Because NGSomics runs entirely on the cloud, it is able to use several different programs in parallel and return data from multiple programs simultaneously.
- **Dockerized**: Docker enables NGSomics to link multiple programs using Docker Compose. Since each program runs as an entirely self-dependent container, it is platform-agnostic and can run on the cloud.

- **Drug correlation**:
- **Intuitive**: NGSomics' front-end interface is easy to use and only requires the input of a few files. It requires no knowledge of command line tools and can be used by any beginner at home.
- **Automatic**: NGSomics' pipeline runs with one click and goes through the process with minimal interruption. At the same time, it allows the user to select certain options and customize the process with ease.
- **Parallelized**: Docker enables NGSomics to run programs in parallel, so that output is returned in the least possible time. It makes use of the cloud's large number of CPU cores and the Docker Compose platform to do so.
- **Easy to visualize**: NGSomics uses iobio and other tools to visualize output with ease in an intuitive format once the pipeline is complete.
- **Affordable**: Since NGSomics runs in the cloud, it requires no processing power and thus the user does not have to purchase machines or clusters to analyze large amounts of data.

How we implemented the different software

Dockerfile example to install dependencies:
https://puu.sh/nGU17/1f7eed6abe.png
Running this in the docker container would allow the freebayes program to be accessible anywhere within the docker volumes, which is based on the local drive.

Testing:

In order to access the quality of our design, NGSomics was compared with other variant calling pipelines using a standardized dataset from GCAT. These tests carried a variety of parameters in order to define in which aspects a software had performed with more effect.

Data Set + Study

NGSomics used e.coli data sets, which contained relatively small amount of reads, to prove the usability of the design, and later transitioned to using the standardized 2009 HG19 genome reference set for comparison tests.
(http://hgdownload.cse.ucsc.edu/downloads.html#human)

Bioplanet GCAT Benchmark Testing:

https://www.one-tab.com/page/Iubz1TFsQLmLP3Ym0mpKWg

Results / Data interpretation:

True Positives (TP) are variants which are correctly identified, whereas **False Positives (FP)** are variants which were incorrectly identified. **True Negatives (TN)** are variants which are neither identified nor present in the organism, whereas **False Negatives (FN)** are variants which are present in the organism but not identified.
The combination of these two measurements to determine the variant caller's accuracy is the **False Discovery Rate**, a ratio of false positives to all variant calls made by the software. The FDR value interprets how many variants are likely to be false positives. **FP / (TP + FP)**
Specificity: True positive rate: ratio of correctly identified variants to total known variants in reference set. Higher sensitivity means it is more likely for a variant in the sample will be identified by the program. **TP / (TP + FN)**
Sensitivity: True negative rate: ratio of non-variant calls to the total number of positions in the reference set that are known to be homozygous with the reference sequence. Inversely related to the number of false positives. **TN / (TN + FP)**

Sensitivity: NGSomics was shown to be more sensitive than the Bowtie2 + Freebayes and Bowtie2 + GATK for this dataset. This means that the NGSomics pipeline had a higher probability of picking up variants within the DNA sequence than the other two software combinations. This increase in sensitivity can be account for as can be seen by the chart that NGSomics calls more indels and SNPs than the other two pipelines.
[insert metadata chart / graph comparing the two with p values]

Specificity: NGSomics was shown to have a slightly lower specificity than the other two pipelines, which although not a significant difference, can be interpreted to mean that NGSomics is more likely to call a variant where there is none in the DNA sequence. However, this error is insignificant, as it only accounts for a **[insert percentage error]** change in specificity.

Positive Variant calls: NGSomics calls more true positives than the other two pipelines, meaning that it does correctly identify more variants than the other two; however, it also calls more false positives, meaning that it also incorrectly identifies more variants than the other two. This elicits the trade-off the user must consider when picking NGSomics; it will pick up more carefully on variants but also increase the chance of an incorrectly identified variant **[insert percentage error]** than other pipeline options.

Negative Variant calls: NGSomics and the other piplines all call nearly the same amount of true negatives, as can be seen by the **[insert percentage error]**, but it also calls fewer false negatives than the other two **[insert cancer]**. This means that NGSomics is less likely to miss areas where a variant is present but not called, which again attests to its higher sensitivity with a tradeoff for lower precision.

Conclusion:

The NGSomics engineering goal was on the whole achieved. Docker dependencies containing necessary genomic analysis functions were executed on the cloud, which contained a Node.js server for a proper UI and mobile interface. The variant calls issued were able to be matched with the results from other genomic analysis software.

Discussion:

Challenges:
Identifying specific parameters to run the script

- When proper parameters per each data set were not specified, the output files could accumulate massive amounts of resources and data, even crashing the server as the function kept running recursively in the background. In addition, indexing files produced from alignment and variant calling slowed down and cluttered up the output data we needed.

- This issue was addressed by inserting a parameter to only retain high confidence variant calls within the final output file. We also implemented a script which removed excess indexing files after the necessary function was performed, as cluttered data files significantly slowed down the operation speed of the program.

Linking Docker Containers to the Cloud

- Linking a localhost or cloud-based volume with the docker virtual machine volumes in order for the programs to communicate effectively with the uploaded data sets was a major difficulty. Furthermore, it was difficult to develop a method to store and tag uploaded files in an accessible and differentiable manner.

Improvements:

Since NGSomics currently only contains capabilities for the interpretation and performance of variant calling, we plan to implement functionality for gene-based protein analysis with WGCNA within the NGSomics pipeline. WGCNA (based on R) would allow the user to find modules based on gene similarities and function in order to provide a complete pipeline for genetics-based analysis. In fact we got so far as to have implemented R-studio with WGCNA installed on the cloud, but ultimately we decided to prioritize variant calling for the NGSomics pipeline.

[insert picture of WGCNA = https://puu.sh/nHUvn/426c62f1b3.png]

Final Implications:

The ever growing need for more detailed and personalized data for genomic medicines underlines the need and reason for NGSomics. A device is needed to branch the large and excessive data files outputted by increasingly extensive application with meaningful interpretations which the common person is able to use and benefit from. Especially when genomic sequencing becomes even more accessible, even to the point where DNA sequencing can be accomplished with merely a mobile device and a probe, such as the DNA sequencer MinION Nanopore, NGSomics becomes more applicable when the user can easily upload the files onto his/her electronic device, **Mobilized genomics with speed and cloud. Dockerized for scalability and accuracy. Intuitive UI for easy user access**

Project Information:
Docker Repository (NGSomics Image): https://hub.docker.com/r/nathan2wong/ngsomics
Github Repository (Files + Backup): https://github.com/nathan2wong/NGSomics
Project Description (Landing Page): http://ngsomics.com
NGSomics Web-app (Workable Demonstration): http://app.ngsomics.com
Bioplanet GCAT report / comparison test: http://bit.ly/ngsomicsreport