Reflective report

Roll number: 9057 | Group 4

Abstract

With the wide application of the next generation sequencing (NGS) technology, there is an increasing demand for analyzing and interpreting big and chaotic NGS data accurately and efficiently. For clinical diagnosis, it is critical to shift from traditional manual interpretation of patient's data to a more intelligent computer-based way. Here, we developed a precision medicine matching system which can call variants from RNA-seq data and match the variants to relevant drugs. The system consists of a database which stores all the information of drugs, a web-based user interface which allows user to input variants and matches the relevant drugs and a sequence analysis workflow which obtains variants from NGS data. As one of the project members, I developed the sequence analysis workflow for the project. Following the VAP pipeline proposed by Adetunji et al, I performed the analysis on RNA-seq data of Hep3B2.1-7 cancer cell line. Total 74205 variants were identified and annotated. All variants were successfully uploaded and a total of 321 drugs were matched. Using our precision medicine matching system, doctors can obtain a list of potential drugs that are personalized for every patient in an effective way. The further development of the project, especially for the sequence analysis, can be designing more accurate and precise variant filtration methods to obtain more confident variants.

Introduction

With the development of sequencing technologies, there has been a shift away from the low throughput sanger sequencing for genome analysis to the next generation sequencing (NGS). Since the NGS is becoming affordable and easy to obtain, there has been an explosion of NGS data from both research and clinics (Metzker, 2010). However, the lack of precise and efficient tools to analyze and provide meaningful biological insights from those genome sequencing data has always been a barrier for clinical diagnosis and research. To remove this barrier, scientists have been developing better sequencing technologies, analysis algorithms and platforms as well as variants to phenotype prediction strategies (Ashley, 2016).

RNA-seq is applicable to numerous research studies, such as gene expression analysis, detection of alternative splicing, allele-specific expression, gene fusions or RNA editing (Wang, Gerstein and Snyder, 2009). Specifically, for variant calling, RNA-seq has some advantages that whole genome sequencing does not possess. RNA-seq can identify mutations at the transcript level, which is closer to the protein level alterations. On the contrary, whole genome sequencing data may provide us with more variants since variants in the non-coding regions are also included, but the majority of them are not passed on to the transcripts and may make the data more chaotic.

We developed the precision medicine matching system to obtain variants from RNA sequencing data and match those variants to approved drugs using the PharmGKB knowledge base. Rather than viewing all the variants, doctors or researchers can upload the sequence analysis results to the matching system and get the relevant drugs and dosing guidelines immediately. The project can be separated into five parts (Figure 1). Sequence analyzer calls and annotates the variants from the RNA sequencing data. The database developer builds a MySQL database which contains all drugs information. The database is

connected with a Java application using Java Database Connection (JDBC) and Gson. Web development builds a web-based user interface using the Model-View-Controller (MVC) framework, servlet, Java server page (JSP), JSTL (JSP Standard Tag Library) and Maven (manage dependencies). The web application allows the users to login in, upload variants, view matching results. The test developer aims to test all the functions of the project, from small to large subsets using Jmeter and Jacoco. The document developer coordinates all team members' work. The software development life cycle we used is the waterfall model since it is a straightforward framework. It is also suitable for a small project like ours. It separates software development into different stages including requirements, design, implantation, test. Specifically, for sequence analysis, the requirement analysis is to identify the overall objective which is to call and annotate variants from RNA-seq data. The design stage is to plan every step of the data analysis pipeline (method). The implementation stage is to use command lines to execute every step (supplementary code). The test stage is to test the function of the sequence analysis workflow (discussion).

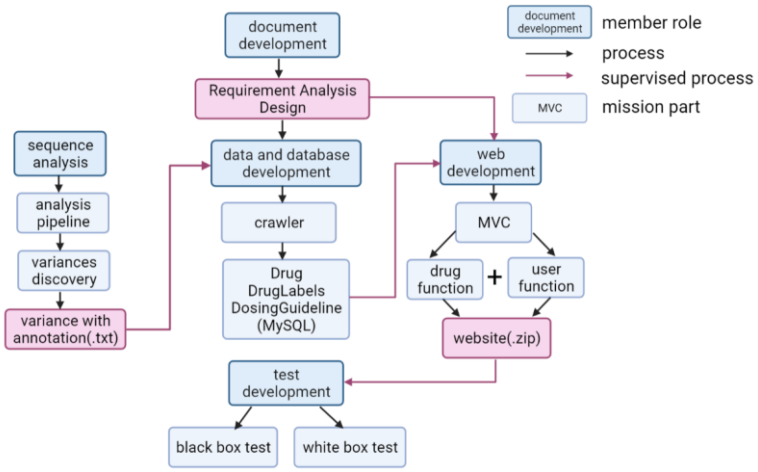


Figure 1. Five parts of the project and their key standards and components (made by the document developer).

The sequence analysis is an initial part of the whole project. It is also critical since the variants may highly affect the drug matching results. A reliable pipeline needs to be chosen as the standard. Several somatic mutation discovery pipelines were taken into consideration (Sheng *et al.*, 2016; Wood *et al.*, 2018; Adetunji *et al.*, 2019). The variant analysis pipeline (VAP) proposed by Adetunji et al was used since its every step is clearly documented. It also compares the performance of different software and uses the highly cited variant caller GATK. It also uses multiple aligners to reduce the false positive rate. The pipeline is tested using RNA-seq data and the results were compared with DNA-seq data to ensure reliability. Moreover, it does not require high server configuration and complicated techniques like machine learning.

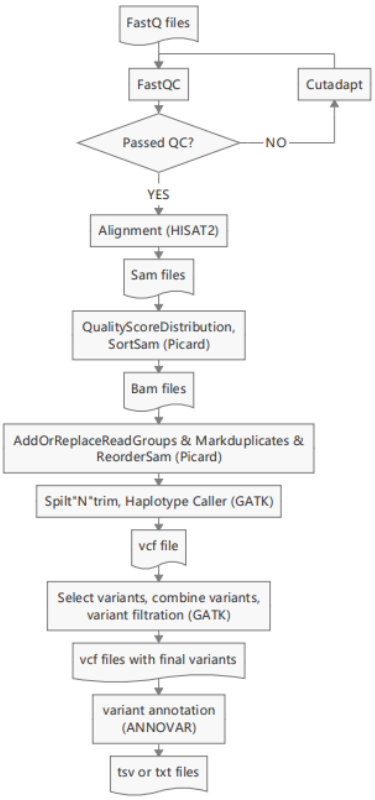


Figure 2. The overall workflow of the sequence analysis pipeline (Own work)

Methods

The standard which sequence analysis follows is the variant analysis pipeline (VAP) proposed by Adetunji et al. The VAP

contains several major steps, including quality control and cleaning of raw sequencing files, alignment and alignment cleaning, variant calling and filtering, variant annotation (Figure 2).

- **Quality control and cleaning of raw sequencing files**

The quality of the raw sequencing files is assessed using FastQC. Then, data cleaning was performed using Cutadapt. If the cleaned FASTQ file passes the quality control, the analysis moves on.

- **Alignment and alignment cleaning**

The reference genome (including FASTA, FAI and DICT files) is downloaded from the GATK resource bundle (Table S4), since using the reference genome provided by the variant calling software gives better performance. The reasons for choosing GRCh38 but not GRCh37 is that GRCh38 is an improved version of the human genome where many gaps were closed, sequencing errors corrected, and centromere sequences modelled.

HISAT2 was chosen to perform the alignment since it is a splice-aware software with more than 4000 citations. A broad-spectrum RNA-seq analysis comparing different software also shows the performance of HISAT2 in variant calling is well (Sahraeian *et al.*, 2017).

After alignment, a SAM file is generated which contains a list of alignment sorting order, read or query name and its related reference sequence name, start and end position and actual sequence (Figure 3). Alignment quality control, sorting and duplicate marking were performed using Picard to make the alignment file ready for variant calling.

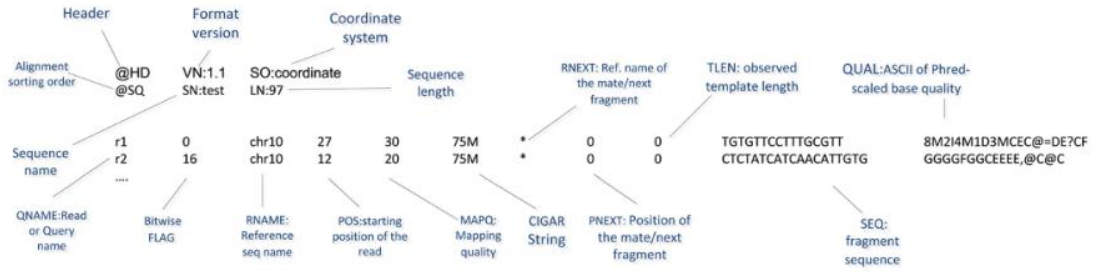


Figure 3. A SAM file example (Pavlopoulos *et al.*, 2013)

Quality Score Distribution It is used to determine the overall 'quality' for the aligned reads. It outputs a figure and table indicating the range of quality scores and the total numbers of bases corresponding to those scores.

Sort Sam This step sorts the input SAM file by coordinate (SO), queryname (QNAME), or some other properties of the SAM record (here sorted by coordinate). Sorting by coordinate is used to streamline data processing and to avoid loading extra alignments into memory. The sort order of a SAM file is found in the SAM file header tag @HD in the field labelled SO.

Add or Replace Read Groups This tool enables the user to replace all read groups in the input file with a single new read group and assign all reads to this read group in the output BAM file. Normally, it is used to give a label to a set of reads that were generated from a single run of a sequencing instrument. Thus, it avoids confusion when emerging the variants. In VAP, this step is used to add the same label to the read groups generated by the same aligner. That is, for instance, the label "hisat" is added to all read groups aligned by HISAT2. Therefore, in the vcf file, the variants are labelled by their aligner's name. These labels are

critical when merging the variants. However, due to technical issue, only HISAT2 is used for alignment in the project. This will be discussed later in the discussion section.

Mark Duplicates

Since the sample failed to pass the quality control of sequence duplication level, mark duplicate is performed to reduce the PCR duplication caused by library construction and optical duplication caused by a single amplification cluster being incorrectly detected as multiple clusters by the Illumina sequencer.

Picard defines duplicated reads as reads aligned to the exact same location, and with exactly the same sequence (Figure 4).

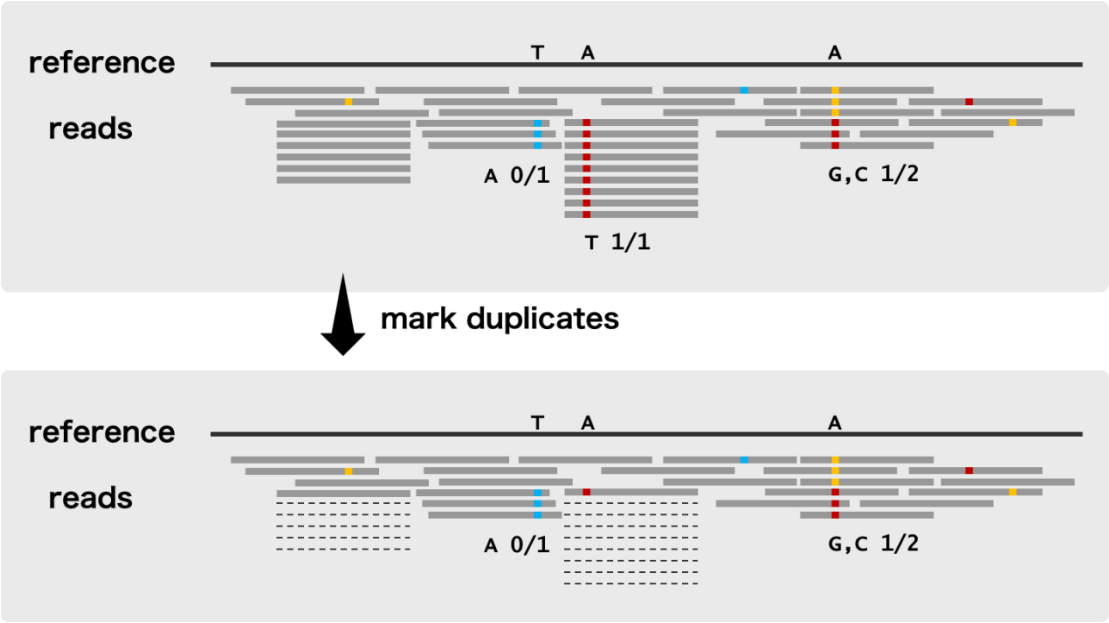


Figure 4. Schematic diagram of mark duplicates (from GATK team). Duplicated reads are reads that are aligned to the exact same location and with the exact same sequence.

Reorder Sam This step sorts a SAM/BAM file with a valid sequence dictionary (DICT file), Reorder Sam reorders reads in a SAM/BAM file to match the contig ordering in a provided reference file, as determined by exact name matching of contigs. Reads mapped to contigs absent in the new reference are dropped.

- **Variant calling and filtration**

Variant calling and filtration were performed using Genome Analysis Toolkit (GATK) (McKenna *et al.*, 2010). Firstly, split N cigar reads is used to split reads that contain Ns in their cigar string (e.g. spanning splicing events in RNA-seq), since the reads that are not spliced properly can cause false positive variants (Figure 5).

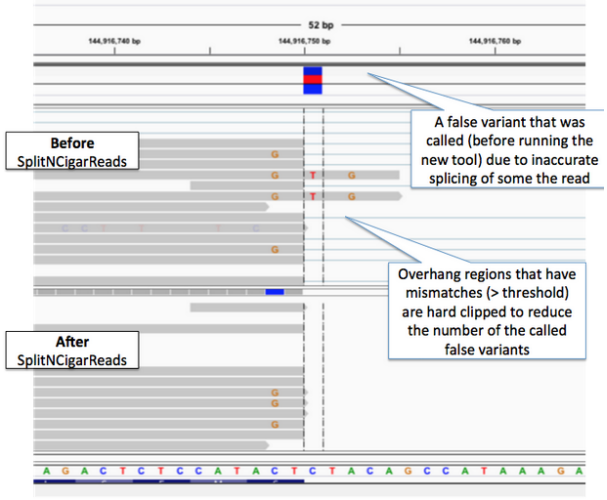


Figure 5. Schematic diagram of split N cigar reads (McKenna *et al.*, 2010). Before spilt N cigar reads, false positive variants may be called.

Haplotype calling is a critical step that calls the variants from the cleaned and sorted BAM files. The haplotype caller is capable of calling SNPs and indels simultaneously via local de-novo assembly of haplotypes in an active region. In other words, whenever the program encounters a region showing signs of variation, it discards the existing mapping information and completely reassembles the reads in that region. This allows the haplotype caller to be more accurate when calling regions that are traditionally difficult to call, for example when they contain different types of variants close to each other (McKenna *et al.*, 2010).

After haplotyping calling, a vcf file is generated which contains the information of variants. The vcf file contains a header and a body. The header includes file format, variant caller version and other annotations (Figure 6a). The body includes information on all variants. For each variant, the chromosome, position, the reference and alternative alleles, the quality score, and some statistics (e.g. BaseQRankSum) are recorded (Figure 6b).

a

```
##fileformat=VCFv4.2
##FILTER=<ID=LowQual,Description="Low quality">
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth (reads with MQ=255 or with bad mates are filtered)">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="Normalized, Phred-scaled likelihoods for genotypes as defined in the VCF specification">
##GATKCommandLine.HaplotypeCaller=<ID=HaplotypeCaller,Version=3.8-1-0-gf15c1c3ef,Date="Mon May 03 02:40:07 CST 2021",Epoch=1619980807242,CommandLineOptions="analysis_t
```

b

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  Label
chr1    14653   .       C       T       1241.77 .       AC=1;AF=0.500;AN=2;BaseQRankSum=1.471;ClippingRankSum=0.000;DP=209;ExcessHet=3.0103;FS=21.134;MLEAC=1;MLEAF=0.50
chr1    15584   .       T       C       321.77  .       AC=1;AF=0.500;AN=2;BaseQRankSum=0.911;ClippingRankSum=0.000;DP=29;ExcessHet=3.0103;FS=1.697;MLEAC=1;MLEAF=0.500;
chr1    15616   .       G       A       26.78   .       AC=1;AF=0.500;AN=2;BaseQRankSum=-1.293;ClippingRankSum=0.000;DP=31;ExcessHet=3.0103;FS=0.000;MLEAC=1;MLEAF=0.500
chr1    15666   .       C       T       263.77  .       AC=1;AF=0.500;AN=2;BaseQRankSum=-0.613;ClippingRankSum=0.000;DP=37;ExcessHet=3.0103;FS=0.000;MLEAC=1;MLEAF=0.500
chr1    15820   .       G       T       667.77  .       AC=1;AF=0.500;AN=2;BaseQRankSum=-0.975;ClippingRankSum=0.000;DP=26;ExcessHet=3.0103;FS=1.752;MLEAC=1;MLEAF=0.500
```

```
.;MLEAF=0.500;MQ=60.00;MQRankSum=0.000;QD=8.12;ReadPosRankSum=-3.722;SOR=1.907      GT:AD:DP:GQ:PL  0/1:107,46:153:99:1270,0,3371
lLEAF=0.500;MQ=60.00;MQRankSum=0.000;QD=11.10;ReadPosRankSum=-1.167;SOR=1.112      GT:AD:DP:GQ:PL  0/1:18,11:29:99:350,0,577
MLEAF=0.500;MQ=60.00;MQRankSum=0.000;QD=0.86;ReadPosRankSum=-1.111;SOR=0.693       GT:AD:DP:GQ:PL  0/1:26,5:31:55:55,0,875
MLEAF=0.500;MQ=60.00;MQRankSum=0.000;QD=7.13;ReadPosRankSum=-0.897;SOR=0.681       GT:AD:DP:GQ:PL  0/1:27,10:37:99:292,0,960
MLEAF=0.500;MQ=60.00;MQRankSum=0.000;QD=25.68;ReadPosRankSum=-3.133;SOR=0.368      GT:AD:DP:GQ:PL  0/1:7,19:26:99:696,0,227
```

Figure 6. The screenshot of the vcf file of Hep3B2.1-7. **a** The header of the vcf file **b** The body of the vcf file. INFO column contains the statistics of the variants.

The select variant is used to select the SNPs from all types of variants. The reason for selecting SNPs is that we are interested in SNPs in this project. Some other types of variants including indels, insertions also affect an individual's phenotype and may cause diseases, but they are not the scope of this project. Next, combine variant is performed to combine variant calling results of different aligners.

Lastly, variant filtration is used to filter out less confident variants. The filtering criteria are as follows (Adetunji *et al.*, 2019):

ReadRankPosSum (RRPS) < -8
Quality by depth (QD) < 5
Read depth (DP) < 10
Fisher's exact test p-value (FS) > 60
Mapping Quality (MQ) < 40
SnpCluster (3 SNPs in 35bp)
Mann-Whitney Rank-Sum (MQRankSum) < -12.5

Some statistics (e.g. FS, MQRankSum) are used to reduce the false positive rate. Some other statistics (e.g. MQ) are used to evaluate the quality of mapping. The filtration of read depth is to exclude the variants where the read depth is very low.

- **Variant annotation**

ANNOVAR is used to annotate the variants (Yang and Wang, 2015). All databases for annotation were downloaded from the ANNOVAR in-build database (Table S4, S6). The annotated variants contain the gene where the variants located, the function of the location (e.g. intronic, UTR3), the cytoband, the variant ID from dbSNP and so on (Table S5). Some columns will be used for matching.

Results

Quality control and preprocessing results

The two paired-end sequencing files failed to pass the quality control of per base sequence content and sequence duplication levels. I solved the first problem by trimming the first 14 bases using Cutadapt (Figure 7). The second problem is treated in the Mark Duplicates step.

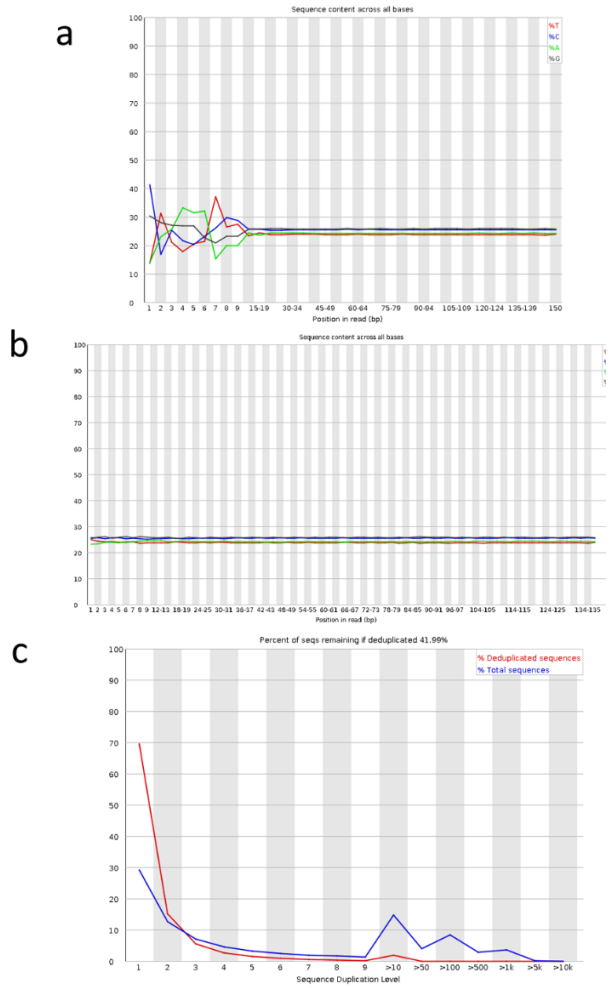
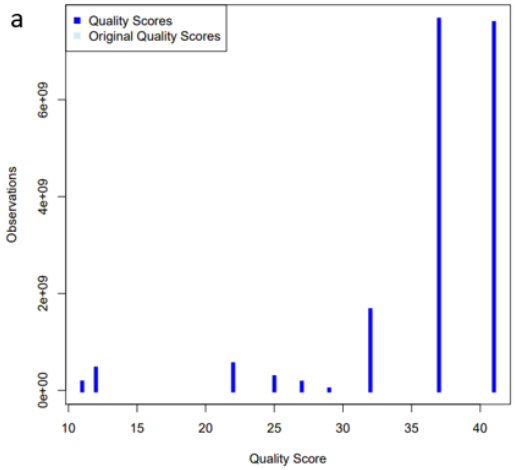


Figure 7. Quality control and data cleaning of raw sequencing data (the results of the first pair-end file is shown). **a** "Per base sequence content" before data cleaning. **b** "Per base sequence content" after trimming the first 14 bases. **c** "Sequence duplication level" before data cleaning.

Quality score distribution

The quality score of the aligned read groups is fine. Most of the bases have a quality score larger than 30 (Figure 8).



b

Quality	Total bases
11	153688511
12	419155257
22	507964003
25	252044858
27	135470657
29	18220037
32	1396507643
37	6815262982
41	7060853709

Figure 8. Quality score distribution of the read groups. The diagram and figure show the number of the bases of different quality scores.

Mapping and mark duplicate statistics

The overall alignment rate of HISAT2 is around a reasonable 95% (Table S2). The duplication rate of the mapped sequence is 31.76% (Table S3).

Variant calling results and its interpretation

A total number of 74205 SNPs passed the filter and were successfully annotated. 95.42% of them are in the intronic, intergenic, UTR, downstream/upstream regions. Most of these SNPs will not alter the protein level or structure. Approximately 5% of the SNPs are located at the exon or/and slicing site (Figure 9). More than 70% of SNPs are transitions. 63373 SNPs are found in the dbSNP database.

Types	Number	Percent
downstream	7308	9.85%
upstream	883	1.19%
upstream; downstream	92	0.12%
exonic	1290	1.74%
splicing	17	0.02%
exonic; splicing	4	0.01%
ncRNA exonic	2083	2.81%
ncRNA intronic	4230	5.70%
ncRNA splicing	6	0.01%
intergenic	6876	9.27%
intronic	33628	45.32%
UTR3	15936	21.48%
UTR5	1839	2.48%
UTR5; UTR3	9	0.01%
Unknown	4	0.01%
Total	74205	/

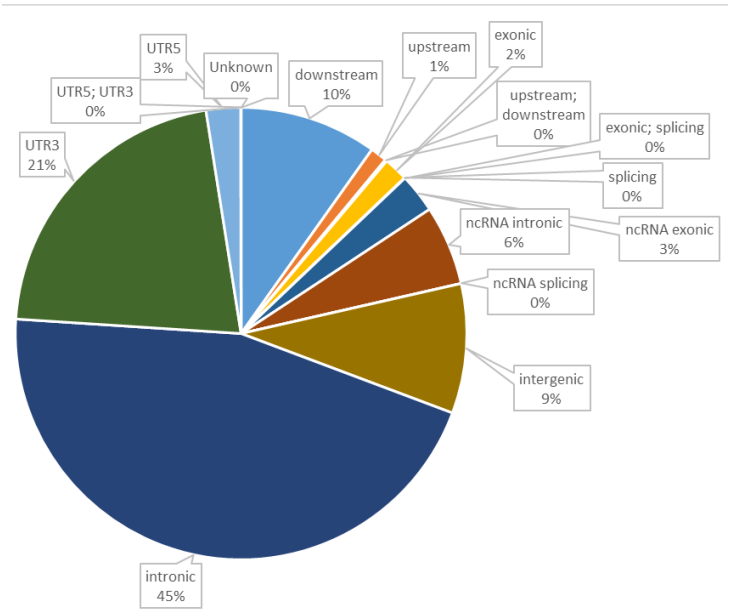


Figure 9. The proportion of different types of variants

Among coding SNPs (SNPs on the exonic or splicing regions), 43.37% are non-synonymous SNPs which affects the peptide sequence and may change protein's function. A few of them are stop gain or stop loss SNPs which may lead to truncated or prolonged peptides.

All 74205 variants were uploaded to the precision medicine match system. Total 321 drugs were matched.

Communicate the sequence analysis result with team members

To let the database and web developers understand how to use sequence analysis results to match the relevant drugs, I explained the sequence analysis pipeline and the meaning of the important columns of the variant annotation file (more details in Table S5).

Provide matching strategies for web developers

I also discussed the matching strategies with web developers. We browsed through the PharmGKB API and found that each drug has a summary markdown that describes the basic pharmacogenetics of the drug. The summary markdown includes the gene name and SNP ID

(from dbSNP) which the drug targets. We planned to use the gene where the variants locate at (Gene.refGene column) and SNP ID (avsnp150 column, planned in the requirement analysis but have not implemented due to time limitation) to search for relevant drugs. If the summary markdown contains the gene name or SNP ID of a certain variant, the drug will be displayed.

Provide suggestions for database development

Since users will upload the sequence analysis results to the matching system, the database should be designed in a way which tailored to the uploaded files. I suggested the database developer that there should be a sample ID column which is used to distinguish the SNP information of different uploaded samples. Also, the sample ID, chromosome (Chr), start position (Start), end position (End), reference allele (Ref), alternative allele (Alt) should all be primary keys (Figure 10), since it may be possible, with very low chance, that two different SNPs are identified on the site of the genome. The pipeline is designed to combine the results of multiple aligners. Therefore, different aligners may generate different alternative alleles on the same site.

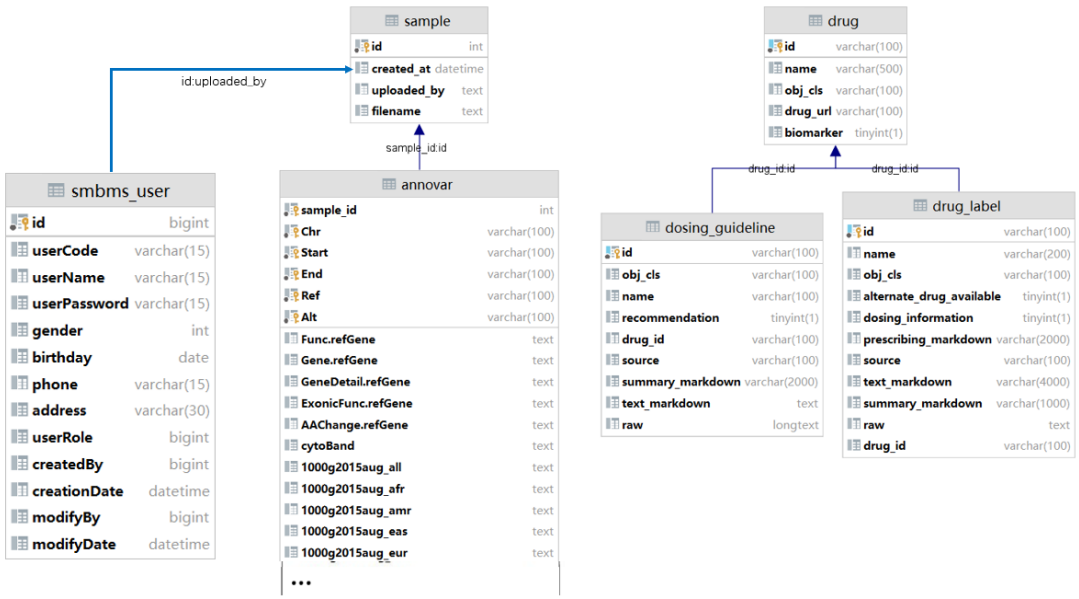


Figure 10. The entity-relationship diagram of the database (made by the database developer). The table annovar stores the information of the input variants.

Other contributions to the group

I participated in every group meeting and brought out my opinions on various issues including requirement analysis, design, the choice of standard (e.g. spring MVC, MVC or Springboot?). I also helped the coder fix some bugs.

After the web and database developer finishes implementing the functions, I ran the source code on my computer and found that there was no error occurred. Previously, before the implementation of the web and database, I also taught two developers how to configure and run the demo system on their IntelliJ IDEA.

Deficiencies of the Perl script of VAP

An error was observed in the original Perl script provided by Adetunji et al (Figure 11). In the

reorder SAM step, the original Perl script uses the FASTA file as reference, while Picard requires the sequence dictionary (DICT). In the method section, I mentioned that reorder SAM reorders the reads according to the contigs. We should provide DICT files that only contains the contig information instead of FASTA files which contains way more information than the required contig information.

```
INFO    2021-04-01 00:08:47     ReorderSam

********** NOTE: Picard's command line syntax is changing.
**********
********** For more information, please see:
********** https://github.com/broadinstitute/picard/wiki/Command-Line-Syntax-Transition-For-Users-(Pre-Transition)
**********
********** The command line looks like this in the new syntax:
**********
**********    ReorderSam -INPUT aln_sorted_mdup.bam -OUTPUT aln_resorted_mdup.bam -REFERENCE /home/yuzj/References/GATK_hg38/hg38.fa -CREATE_INDEX TRUE
**********


ERROR: Unrecognized option: REFERENCE

USAGE: ReorderSam [options]

Documentation: http://broadinstitute.github.io/picard/command-line-overview.html#ReorderSam

Not to be confused with SortSam which sorts a SAM or BAM file with a valid sequence dictionary, ReorderSam reorders
reads in a SAM/BAM file to match the contig ordering in a provided reference file, as determined by exact name matching
of contigs.  Reads mapped to contigs absent in the new reference are unmapped. Runs substantially faster if the input is
an indexed BAM file.
Example

java -jar picard.jar ReorderSam \
INPUT=sample.bam \
OUTPUT=reordered.bam \
SEQUENCE_DICTIONARY=reference_with_different_order.dict
```

Figure 11. The error message when running the original Perl script.

Another deficiency is the Perl script does not exit after an error occurs. It simply continues running which wastes the resources. Adding an exit function which allows the program to exit if an error occurs is necessary. Also, it is better to add a configuration step before the start of the whole pipeline. In the configuration step, the program can check whether all required reference files and software have been appropriately placed or installed.

Discussion

In this project, we developed the precision medicine matching system which calls variants from RNA-seq data and matches those variants to drugs. Our sequence analysis workflow, web user interface and database together can achieve basic functions of precision medicine.

Throughout the project, we followed the waterfall model. In the aspect of the big project, we first had a whole requirement analysis and then a more detailed design analysis. Then, the document developer allocated the job to all team members to finish the implementation. Finally, the test developer test all functions of the project. In the aspect of my own task-sequence analysis, I also followed the waterfall model, where I first confirmed the objectives and then designed the specific steps, after that, I executed all steps using server, finally tester and I tested the workflow.

In the project, NGS was used to sequence the cancer cell line mRNA. One may ask whether it is better to use third-generation sequencing. One of the advantages of third-generation sequencing is that there is no need to do PCR amplification before sequencing. For NGS data, PCR amplification is often required to increase the library size. Thus, it introduces the PCR duplication which may affect the read depth in some regions. This may influence the variant calling results. However, the third-generation sequencing is also accompanied by a high error rate, which may also affect variant calling results (Pei *et al.*, 2020).

The data I analyzed is the mRNA sequencing data. However, approximately 20% of variants are not on the mRNA. This may be caused by the limitations of the library preparation and

data analysis. The isolated mRNA may contain other types of RNA. Also, it is possible that some reads are mapped to non-coding regions since the whole genome not transcripts is used as reference.

One of the limitations is that I did not use multi aligners to perform the alignment. Adetunji et al propose that using multi aligners (TOPHAT, HISAT2, STAR) and selecting the intersect variants can reduce the false positive rate. However, HISAT2 is an upgraded version of TOPHAT (Marx, 2020). TOPHAT users are suggested to migrate to HISAT2. I think it is not so meaningful to merge the results generated by TOPHAT and HISAT2. However, combining HISAT2 with STAR is a good way to reduce false positive variants. Further development of the sequence analysis workflow can consider using both HISAT2 and STAR.

Another limitation is the lack of test on the sequence analysis pipeline. The test developer and I did not discuss much on testing the sequence analysis pipeline. Only the Hep3B2.1-7 RNA-seq data was used to test the pipeline. We should collect more amount and types of data and test the pipeline more thoroughly in further development.

The third limitation is the inaccuracy of the drug matching. However, due to the limited information provided in the PharmGKB database, more information is required to make more accurate matching. For instance, a more detailed target description of the drugs in the drug labels may improve the matching accuracy.

Our project can be further developed in several aspects. Firstly, the data that I analyzed is the somatic mutations from a cancer cell line. However, in clinical diagnosis, it is possible that some data comes from germline. Germline mutation is different from somatic mutation since germline mutation can be inherited. These mutations need to be analyzed in the background of a family. The system can be further developed to support uploading the variant files of the whole family. The system can compare variants from different family members and identify whether a certain mutation is sporadic or familial. Secondly, the number of variants is still too high. the majority of the variants are not the driver for cancer. They are simply passager mutations which cause by the instability of the cancer genome. However, finding the true “driver” is difficult and cannot be done using only variant filtration software. Combining the biological discoveries with the variant calling results is necessary.

As for the collaboration between other group members, what I did well is I provided the web and database developers with the first version of the sequence analysis results at an early date. This gives them more time to look through the data and understand the format of the data. However, I did not explain the data at that time, which caused some confusion. The web and database developers misunderstood my point, thinking that the database should include all columns of the variant annotation files. However, what I meant is they could choose the necessary columns. Next time I will provide the data along with the documentation.

References

Adetunji, M. O. *et al.* (2019) ‘Variant analysis pipeline for accurate detection of genomic variants from transcriptome sequencing data.’, *PloS one*, 14(9), p. e0216838. doi: 10.1371/journal.pone.0216838.

Ashley, E. A. (2016) ‘Towards precision medicine’, *Nature Reviews Genetics*, 17(9), pp. 507–522. doi: 10.1038/nrg.2016.86.

Marx, V. (2020) 'Bench pressing with genomics benchmarkers', *Nature Methods*, 17(3), pp. 255–258. doi: 10.1038/s41592-020-0768-1.

McKenna, A. *et al.* (2010) 'The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data.', *Genome research*, 20(9), pp. 1297–1303. doi: 10.1101/gr.107524.110.

Metzker, M. L. (2010) 'Sequencing technologies — the next generation', *Nature Reviews Genetics*, 11(1), pp. 31–46. doi: 10.1038/nrg2626.

Pavlopoulos, G. A. *et al.* (2013) 'Unraveling genomic variation from next generation sequencing data.', *BioData mining*, 6(1), p. 13. doi: 10.1186/1756-0381-6-13.

Pei, S. *et al.* (2020) 'Benchmarking variant callers in next-generation and third-generation sequencing analysis', *Briefings in Bioinformatics*. doi: 10.1093/bib/bbaa148.

Sahraeian, S. M. E. *et al.* (2017) 'Gaining comprehensive biological insight into the transcriptome by performing a broad-spectrum RNA-seq analysis.', *Nature communications*, 8(1), p. 59. doi: 10.1038/s41467-017-00050-4.

Sheng, Q. *et al.* (2016) 'Practicability of detecting somatic point mutation from RNA high throughput sequencing data.', *Genomics*, 107(5), pp. 163–169. doi: 10.1016/j.ygeno.2016.03.006.

Wang, Z., Gerstein, M. and Snyder, M. (2009) 'RNA-Seq: a revolutionary tool for transcriptomics', *Nature Reviews Genetics*, 10(1), pp. 57–63. doi: 10.1038/nrg2484.

Wood, D. E. *et al.* (2018) 'A machine learning approach for somatic mutation discovery', *Science Translational Medicine*, 10(457), p. eaar7939. doi: 10.1126/scitranslmed.aar7939.

Yang, H. and Wang, K. (2015) 'Genomic variant annotation and prioritization with ANNOVAR and wANNOVAR', *Nature Protocols*, 10(10), pp. 1556–1566. doi: 10.1038/nprot.2015.105.

Supplementary information

Table S1. Software/language version

Software/language name	Version
Perl	5.20.2
FastQC	0.11.9
Cutadapt	3.2 with Python 3.6.6
HISAT2	2.1.0
Picard	2.25.1
GATK	3.8-1-0-gf15c1c3ef
ANNOVAR	2018 Apr 16

Table S2. Mapping statistics

Sample	Total number of reads	aligned concordantly 0 times	aligned concordantly exactly 1 time	aligned concordantly >1 times	overall alignment rate
Hep3B2-1-7	61615015	6001851 (9.74%)	50773244 (82.40%)	4839920 (7.86%)	95.17%

Table S3. Mark duplicate statistics

LIBRARY	Unpaired reads examined	Read pairs examined	Secondary or supplementary rds	Unmapped reads	Unpaired read duplicates
Label	3173562	57052138	17841133	5952192	2424369
LIBRARY	Read pair duplicates	Read pair optical duplicates	Percent duplication	Estimated library size	
Label	17408850	428098	0.317554	74401456	

Table S4. The source of the reference genome and databases

Name	Extension	FTP/command line
Reference genome	FASTA	ftp://gsapubftp-anonymous@ftp.broadinstitute.org/bundle/hg38/*
Reference genome index	FAI	
Sequence dictionary	DICT	
ANNOVAR database	TXT	annotate_variation.pl -buildver hg38 -downdb -webfrom annovar database_name humandb

Table S5. Explanation of some important columns in variant annotation files

Column name	Explanation
Chr	Chromosome
Start	Start site on the chromosome
End	End site on the chromosome
Ref	Reference allele
Alt	Alternative allele
Func.refGene	Annotate the region where the mutation site is exonic, splicing, exonic; splicing, UTR5, UTR3, intronic, ncRNA exonic, ncRNA intronic, ncRNA splicing, upstream, downstream, intergenic. Here exonic only represents coding exonic portion, not include UTR3 and UTR5. When a mutation is located in both UTR3 of one gene and UTR5 of another gene, the column outputs "UTR5,UTR3". When a mutation will start in one gene and downstream in another gene, the column exports "downstream, downstream".
Gene.refGene	The name of the gene associated with the variant
GeneDetail.refGene	Describe the variation of UTR, splicing, ncRNA splicing or intergenic regions. When the value of Func.refGene column is exonic, ncRNA exonic, intronic, ncRNA intronic, upstream, downstream, upstream; downstream, ncRNA_UTR3, ncRNA_UTR5, this column is empty; when the value of Func.refGene column is exonic; splicing, it means that the

	site is located at some exonic regions of the transcript, and the splicing region of other transcripts. In this case, GeneDetail will give the effect of the site on the transcript splicing, for example, NM_172210:exon6:c.1090+5C>A, which means the mutation located on the transcript NM_172210, exon 6 indicates the sixth human exon, c.1090+5C>A indicates that a mutation from C to A occurred 5 bp downstream of 1090 bp of the cDNA; when the value of the Func.refGene column is intergenic, the format of this column is dist=1366; dist=22344, which means the distance between the mutation site and the genes on both sides.
ExonicFunc.refGene	SNV or indel mutation types in exon regions (SNV mutation types include synonymous SNV, missense SNV, stop gain SNV, stop loss SNV and unknown; Indel mutation types include frameshift insertion, frameshift deletion, stop gain, stoploss, non-frameshift insertion, non-frameshift deletion and unknown)
AAChange.refGene	Amino acid changes, only when Func.refGene is listed as exonic or exonic; splicing, the column will have a result. Annotate according to each transcript (for example, AIM1L:NM_001039775:exon2:c.C2768T:p.P923L, where AIM1L represents the name of the gene where the variation is located, NM_001039775 represents the transcript ID where the variation is located, and exon2 indicates that the variation is located In the second exon of the transcript, c.C2768T indicates that the mutation causes the cDNA to change from C to T at position 2768, and p.P923L indicates that the mutation causes the amino acid at position 923 of the protein sequence to change from Pro to T. Leu)
cytoBand	The chromosome segment where the mutation site is located (observed by Giemas staining)
avsnp150	dbSNP ID

Table S6. All databases used to annotate vcf files

hg38_AFR.sites.2015_08.txt, hg38_ALL.sites.2015_08.txt, hg38_AMR.sites.2015_08.txt, hg38_avsnp150.txt, hg38_clinvar_20180603.txt, hg38_cosmic70.txt, hg38_cytoBand.txt, hg38_dbnsfp35a.txt, hg38_EAS.sites.2015_08.txt, hg38_esp6500siv2_aa.txt, hg38_esp6500siv2_all.txt, hg38_esp6500siv2_ea.txt, hg38_EUR.sites.2015_08.txt, hg38_exac03.txt, hg38_gnomad_exome.txt, hg38_gnomad_genome.txt, hg38_icgc28.txt, hg38_intervar_20180118.txt, hg38_refGeneMrna.fa, hg38_refGene.txt, hg38_refGeneVersion.txt, hg38_SAS.sites.2015_08.txt

Code and data availability

All command line information, vcf files and annotated variant files are submitted along with the report.