Computer Assignment 8: Characterizing Genetic Variants from Next-generation Sequencing Data

**1.1** Why are there two *fastq* files, one with the \_1.fastq extension and another with \_2.fastq? (Hint: this has to do with the way that reads are made in the NGS technique used here).

The reads are made by sequencing the sequence in more than one run. In this case, since there are two fastq files, this suggests that the NGS technique involved running the sequencing twice.

**1.2** Phred scores above 30 are generally considered trustworthy. However, 99.9% accuracy across millions of base pair calls still results in thousands of sequencing errors. How can we accurately identify variations in our genome (compared to the refence genome) and distinguish these from simple sequencing errors? (Hint: remember the highly parallelized approach of NGS.)

The power in the NGS approach is that it sequences our base pairs multiple times, and so if we align these sequencings, even if individually a single sequence may contain errors, we can cross reference the sequences that all are supposed to represent the same variation of the genome, and figure out what the “true” base pair should be.

**2.1** What is the purpose of all of the ‘echo’ commands in the script? If you’re unsure, then consider the results of the command in your terminal: echo Hello World >> log.txt

Consider when you press hello world into the command line

It would return an error, because it is expecting that as an input.

However, if you press echo hello world, the command line now knows to interpet that as saying “direct what ever follows from this ‘echo’ into the stdout”, where the default stdout is the command line interface.  
  
Now, lets look at if we were to do:

%%%[`date`]%%% Running BWA alignment on fastq 1... >> $logfile 2>&1

The bash script would be confused, because rather than expecting whatever to the left of the >> as something to write into the logfile2, it thinks it is just saying, let’s run this as an input/command. Rather, what we want to do is say, “hey, here is this text, and i want you to redirect this text into our stdout”. Echo allows us to redirect text into our intended stdout, and in this case, that stdout is whatever file is after the >>.

**2.2** There are two flags specified for the bwa aln commands, and one flag for the bwa sampe command. Check the [BWA manual](http://www.htslib.org/doc/samtools.html) and describe what these parameters are specifying. (A quick Google search can often turn up more useful explanations than program documentation).

For bwa aln, there are two flags used in the following sample command:

/project2/bios26120/Lab8/software/bwa/bwa aln -q 5 -t 28 $reference $fastq2 > ${prefix}\_2.sai #2>> $logfile

The two flags are -q, and -t.

1. The -q flag takes in an int parameter for read trimming, meaning the BWA will trim a read down to the size of the int parameter, and in our case, that means it will trim it down to 5 reads.
2. The -t flag takes in an int parameter for the number of threads, and suggests that BWA can support multi-threaded mode. In this case, we are running 28 threads simultaneously.

For bwa sampe, there is one flag used in the following sample command:

/project2/bios26120/Lab8/software/bwa/bwa sampe -P $reference ${prefix}\_1.sai ${prefix}\_2.sai $fastq1 $fastq2 2>> $logfile

The flag used here is -P

1. The -P flag tells BWA to load the entire FM-index into memory to reduce the number of disk operations needed. This is helpful for when the size of the genome is large, and thus is time efficient to do this.

**2.3** Look into SRS003660.align.log with *less* command. About how long did the alignment of each *fastq* file take? How long did the *bwa sampe* process take to complete?

1. The BWA alignment on the 1st fastq file took 1 minute and 41 seconds
2. The BWA alignment on the 2nd fastq file took 1 minute and 48 seconds
3. The bwa sampe process took 17 minutes and 40 seconds

**2.4** List the names the files generated by this run. Which is the largest, and how large is this file? *(-lh* option of *ls* converts bytes to larger metric units).

The largest of these files is the SRS003660.bam file, and this file is 1.3GB

**2.5** How many sequences were processed in total from both paired-end reads? How many of these reads were rejected by quality control (QC)?

8526004

There were 17,052,008 reads that were processed in total from both pair-end reads. There were ZERO reads that were rejected by the quality control

**2.6** What percentage of these reads mapped to the reference genome in your alignment process?

94.16% of the reads mapped to the reference genome in my alignment process.

**2.7** Each short read in our NGS data has a mate that is taken from the same segment of DNA but in the opposite direction. We could reasonably expect, then, that any given read and its pair will map to the same chromosome. How many paired-end reads in total (irrespective of mapQ) did map to different chromosomes? Why might this occur?

78,846 pair-end reads in total mapped to a different chromosome. This may happen because of translocation, meaning that some part of the chromosome is transferred to another chromosome, and as a result they map to different chromosomes.

**3.1** For example, let’s say we have 36 reads that map to nucleotide position 104,204 of chromosome 10. 28 of these reads place an “A” base at this position, and 8 reads have a “T”.

**a)** Intuitively, would you conclude that the individual has a heterozygous AT genotype? Or would you conclude that this individual has a homozygous AA genotype, and the eight “T” reads are sequencing or alignment errors?

On the one hand, if the individual has a heterozygous A genotype, maybe we would expect more of a 50-50 proportion between nucleotides that map to “A” at position 104,204, and nucleotides that map to “T” at position 104,204. On the other hand, it seems very improbable that literally 8 out of 36 times, an alignment error would occur, as alignment errors should probably be pretty low or else genetics would be so random that we can’t even say that we’re the same species. However, we might also consider that these alignment errors are due to both insertion deletion errors, as well as single nucleotide polymorphisms, (since we are still reading at the same position, insertions or deletions “shift” our index, which can lead to a different read). Then, to me it seems more probable that this individual simply has a homozygous AA genotype, but that eight of these ‘T’ reads are just alignment errors.

**b)** How would the knowledge that the sequencing or alignment quality scores on the reads that produced the eight “T” calls are very low affect your confidence in your conclusion?

If we know that the sequencing or alignment quality scores on the reads that produced the eight “T” calls are very low, that suggests that it is highly probable that this position have alignment errors, and thus it would increase the confidence in the conclusion I have described above.

**c)** If the individual is truly AT heterozygous, how many of the reads (on average) would you expect to indicate a T base at this position?

If the individual is truly AT heterozygous, we would probably expect on average that half of the bases are T’s, or 18.

**3.2** Which parameter should you change to limit the genotyping to chromosome 5? Open ***mpileup\_genotyping.sh*** with nano and change this parameter accordingly (change it to chr5). Save your modified script. Hint: consult the ***mpileup*** section of the [*samtools*](https://www.htslib.org/doc/samtools.html) manual

The line we are interest in is:

/project2/bios11141/CompAsn8/software/samtools/samtools mpileup -r chr5 -t SP -uv -f $reference $inbam | /project2/bios11141/CompAsn8/software/samtools/bcftools call -mv > ${prefix}.raw.vcf

And here, we see that we just needed to change the string following the -r flag chrome chr7 into chr5.

**3.3** One crafty student suggests that to save further computational time, we could have limited our **alignment step** to align our reads solely to chromosome 6 rather than the entire genome. Why might this not be a good idea?

If we limit our alignment step to align our reads solely to chromosome 6, then we are completely disregarding information that we care about. Yes, there is less computation, but by this argument, we could just not align anything at all.

**3.4** Look into SRS003660.genotype.log with *less*. Which step of this process took the longest to complete?

The step of this process that took the longest is creating the raw vcf file.

**3.5** Output the last 8 lines of your vcf file with the *tail* command. List the mutations from your data (reference nucleotide, variant nucleotide, and chromosomal coordinates) that pass this Phred quality cutoff.

| Chromosomal Coords | Reference nucleotide | Variant nucleotide | Phred value (>30) |
| --- | --- | --- | --- |
| 181114707 | C | T | 61 |
| 181237105 | C | G | 33.41 |
| 181243433 | T | G | 81 |
| 181259957 | C | T | 55 |
| 181260211 | C | T | 85 |

**3.6** Approximately how many variants are in our ***.vcf*** file output? Use the *wc* command with the required parameter to give the number of lines (check *man wc*) in the ***.vcf*** file. Header lines start with “#” and are not variants (you can use *grep “#” SRS003660.flt.vcf | wc* with the proper parameter to find the number of header lines).

The number of print\_new lines in our flt.vcf file is 7334, but 3395 of these lines are header file lines, and so there are 3939 lines, and since each variant is a line, there are 3939 variants.

**3.7** Approximately how many high-quality variants are in your ***.hq.vcf*** output file?

There are now 6507 new\_lines in the new file, and since yet again there are 3395 lines that are header files, there are 3112 high quality variants

**4.1** From this output, how many SNPs and how many indels were present in your ***hq.vcf*** file?

According to this output, there were 2914 single nucleotide polymorphisms, and 198 indels (insertions or deletions).

**4.2** Does ANNOVAR report anything concerning of which we should be mindful with regards to our ***hq.vcf*** input?

ANNOVAR gives us a warning for the refGene annotation, describing that a total of 515 sequences were ignored because of a lack of correct ORF annotations.

**4.3** What are four types (choose any four) of non-exonic functional regions from which we see annotated variants in our results file (this information is in the sixth column)?

Other than exonic functional regions, there are also intronic, UTR3, UTR5, intergenic, ncRNA\_intronic functional regions.

**4.4** How many of our annotated variants are exonic? (Remember to exclude the header line from the line count!). Why are we mainly interested in exonic variants?

There are 614 variants that are exonic. We are mainly concerned with exons, because exons that the regions that are actually “expressed”, and so the mutations (whether SNPs or indels) in these regions would actually actualize into proteins and phenotypes that we are concerned about.

**4.5** Construct another set of commands to create a third file that contains all nonsynonymous exonic variants. Paste the commands you used in your lab text. How many of the exonic variants are nonsynonymous amino acid substitutions? (NOTE: a query for “synonymous” would pick up all lines that contain synonymous, even if that word is nested within a larger word such as nonsynonymous. **Be careful of this fact and always verify that your *grep* search is picking up exclusively those lines you intend!**

Out of these 614 variants that are exonic, 255 of them are nonsynonymous. Here is the command I used to find this:

head -1 SRS003660.chr5.annovar.hg38\_multianno.txt > SRS003660.chr5.annovar.hg38\_multianno.nonsynonymous.exonic.txt

grep 'nonsynonymous' SRS003660.chr5.annovar.hg38\_multianno.exonic.txt >> SRS003660.chr5.annovar.hg38\_multianno.nonsynonymous.exonic.txt

SRS003660.chr5.annovar.hg38\_multianno.nonsynonymous.exonic.txt | wc

**4.6** How many of the exonic variants are insertions? How many are deletions?

Hint: if you search for ‘insertions”, “deletions” or “frameshift” using grep and the output you get is 0 then open up the output file using Excel and look at the columns that correspond to the nucleotides. Think about what the corresponding mutation should be when there is a gap '-' in different columns and you can count the number of insertions and deletions by counting the number of gaps in each column.

I could not figure this out using grep, my idea was to first use grep to get all the nonsynoymous mutations, then use awk to find those rows whose 5th delimited column contains a ‘-’ (which implies gap) to get the number of deletions, and then also subtract the total number of nonsynonymous from this to get the number of insertions. Here is the following command I used:

grep 'nonsynonymous' SRS003660.chr5.annovar.hg38\_multianno.exonic.txt | awk '$5 == -' | wc

But unfortunately I don’t really know how to use awk so I didn’t know how to get it to actually check for the -, and so it returned 0s instead.

UPDATE:

I just counted the number of gaps in each column in the excel file, and found that there are:

8 deletions, and 0 insertions.

**5.1** There should be a few exonic variant loci you discovered that are associated with a pathology through genome-wide association studies (GWAS) in the ClinVar database. Identify a non-synonymous mutation that is associated with a specified condition. Which gene does this variant occur in, and with what phenotypic trait does it correlate? Hint: search the variant in the NCBI data base:<https://www.ncbi.nlm.nih.gov>

The non-synonymous mutation I chose was the A>G dbSNP, located at position 7870860, and has dbSNP reference ID of rs1801394. It is a mutation that can occur in two genes: the first is a missense at MTRR (5-methyltetrahydrofolate-homocysteine methyltransferase reductase (plus strand)), and the second is a 2KB Upstream variant occurs in the FASTKD3 gene (FAST kinase-domain 3 (minus strand)). The phenotype that this mutation results in includes: benign phenotypes such as, Disorders of Intracellular Cobalamin Metabolism, or Homocystinuria-Megaloblastic anemia due to defect in cobalamin metabolism, cblE complementation type, or Risk-Factor phenotypes such as susceptibility to Neural tube defects, folate-sensitive, or susceptibility to Down Syndrome.

**5.2** What are the relevant nucleotide and amino acid substitutions for this non-synonymous mutation (hint: look at columns D, E, and J)?

The relevant nucleotide and amino acid substitution for this non-synonymous mutation is a switch from A to G nucleotide, resulting in a substitution into amino acid M, or Methionine.

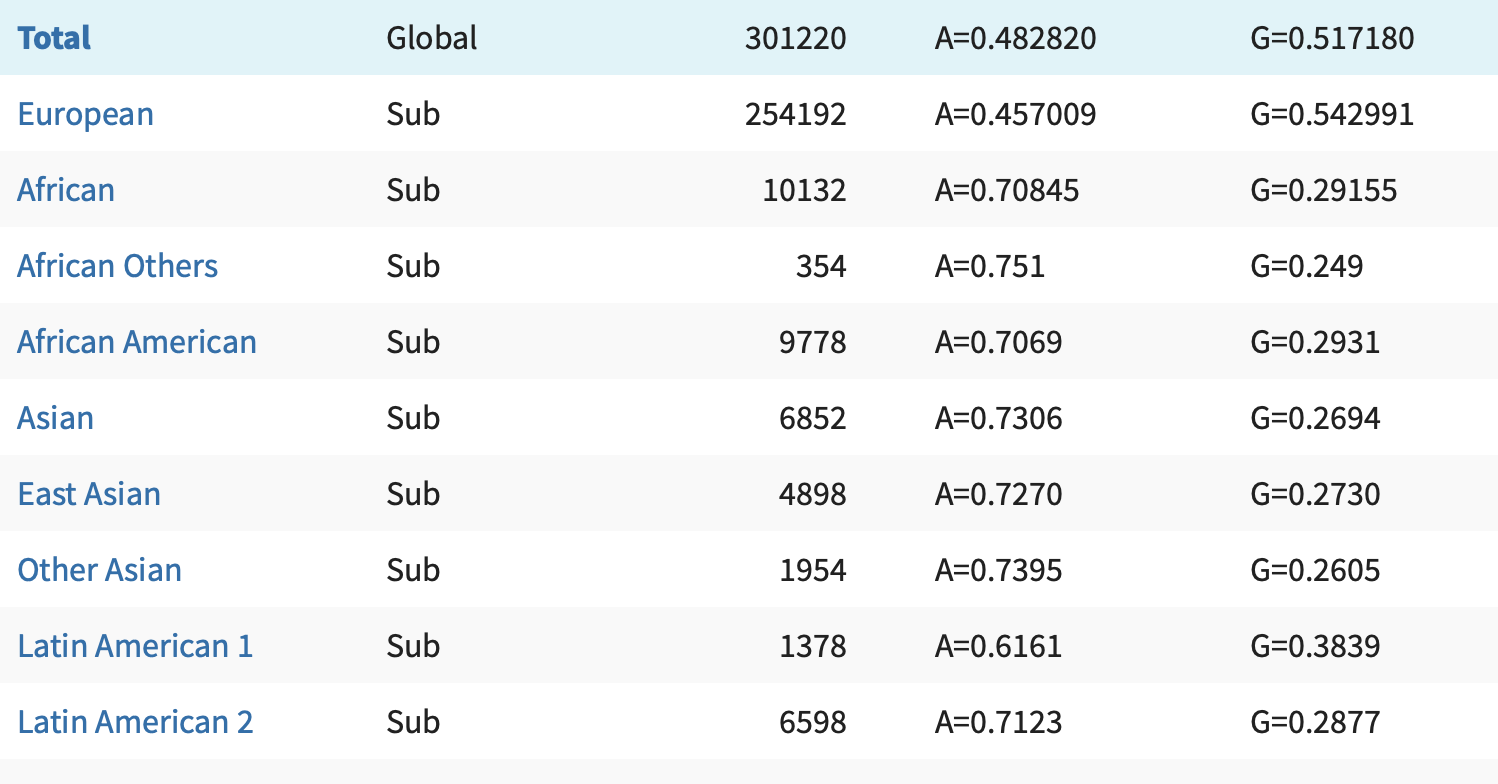
**5.3** This allele has a ***dbSNP*** reference ID, listed in the ***avsnp150*** column. This means that this variant has been discovered previously. Navigate to the NCBI ***dbSNP*** (<http://www.ncbi.nlm.nih.gov/SNP/>) database and search for information on this SNP. What is the ancestral allele in human populations at this site? Which is the minor allele, and what is its allele frequency (minor allele frequency: MAF) for the different projects listed?

The ancestral allele in human population at this site is A. There are a whole bunch of allele frequencies listed, and it actually is unclear which one is the minor allele, since the frequencies are quite competing. I included a screenshot below to illustrate this:



**5.4** Navigate to the population diversity section. Are there any population groups that noticeably differ in their genotype frequencies at this position according to this data? What is the range of MAF across these populations? You can find the data in the frequency tab boxed in the screenshot below.

Looking at the population diversity section, we notice that the total distribution between the two alleles is almost 50/50. However a lot of the African, Asian and Latin Americans have a smaller MAF, ranging from 0.249 to 0.384 frequency ratio for the G. Meanwhile, Europeans have a much greater minor allele frequency, with the ratio being 0.542991, making it actually the “major” allele. I have attacted a screenshot below for reference:



**5.5** There are many different metrics used by dbNSFP

<http://varianttools.sourceforge.net/Annotation/dbNSFP>

to try to predict the effects of our non-synonymous variants, such as SIFT, PolyPhen-2, etc. Pick two metrics to research and briefly summharize what kinds of criteria are considered in these metrics’ scoring schemes. For your chosen variant, describe what these metrics predict about any functional ramifications.

The two metric scoring systems I will choose are: SIFT and PolyPhen-2.

For the SIFT score, the smaller the number, the more damaging it is, while the larger the score, the more tolerated it is. It also ranges from 0 to 1. The score for our allele is 0.002,. Which is below the threshold (0.05), meaning that our SNP is damaging.

For the PolyPhen\_HDIV\_2 score, its the opposite, and its based off of HumDiv, the hdiv\_prob. It is also scored on a scale from 0 - 1, where 0.957 is the threshold for probably damaging. Since our PolyPhen-2 score is 1, this suggests that our SNP is probably damaging.