NGS Data Analysis Workshop

Instructor: Lixing Yang, Ph.D.
Ben May Department of Cancer Research
lixingyang@uchicago.edu

Prerequisites:

Operating System: MacOS, Linux (Ubuntu, Fedora, etc.), or Windows with Cygwin installed to mimic a Linux environment.

Java: JRE v1.6 or above.

Linux skill: Familiar with entry level Linux commands: ls, cd, cat, grep, sort, uniq, wc, cut, more, less.

Statistics: Basic knowledge, R installed, basic knowledge of R.

Tutorial:

This tutorial is to demonstrate the analysis steps from getting the raw sequencing data from service provider to delivering actionable mutations to physicians. The sequencing data are from colorectal cancer patients.

0. Preparation

0.a. Data. All data and required software packages are in the ngs_workshop.zip file. Unzip the file. There is an empty folder called /results. Put all the output files there. Try to avoid putting the output files and input files in the same folder. There is a potential risk to remove or overwrite the raw data files. Always write to a different folder and create symbolic links (`ln -s`) pointing to the raw data if necessary.
PC users, copy the ngs_workshop folder under Cygwin installation folder. For example, if your Cygwin is installed at D:\cygwin, you can copy the ngs_workshop folder to D:\cygwin\home\username. Once you open the Cygwin terminal for the first time, a home folder will be created.

0.b. Terminal. Open a terminal, change your current path to ngs_workshop. The entire tutorial assumes your current path is ngs_workshop folder. Otherwise, you will need to change the paths of input, output, reference files in order to run the codes correctly. You can use `pwd` to find out your current path. All tasks in this tutorial are to be done in terminal by default.
PC users, you can use "Shift+Ins" to paste texts into the command line window.

1. Sample QC

1.a. Basic practice: sequencing quality (FastQC).
Exercise 1: Input data can be found in /data/1.QC. Are these two datasets of high quality?

1.a.1. Launch FastQC:
Mac/Linux users,

```
$ ./tools/FastQC/fastqc
```

PC users, in windows file explorer rather than Cygwin terminal, double click run_fastqc.bat in the FastQC folder.

1.a.2. Load both files and speculate the quality matrices.

1.b. Advanced practice: sample swapping (NGSCheckMate).

2. Read mapping

2.a. Basic practice: standard pipeline (bwa, hg19) and alignment QC.

Exercise 2: Align the reads by BWA. Input data can be found in /data/2.mapping.

2.a.1. Index the reference genome:

```
$ ./tools/bwa-0.7.15/bwa index ref/hg19.fa
```

2.a.2. Align the reads:

```
$ ./tools/bwa-0.7.15/bwa mem ref/hg19.fa data/2.mapping/TCGA-AA-A01T.tumor.1.fq.gz data/2.mapping/TCGA-AA-A01T.tumor.2.fq.gz | gzip -3 > results/TCGA-AA-A01T.tumor.sam.gz
```

2.a.3. Index the reference genome:

```
$ samtools faidx ref/hg19.fa
```

2.a.4. Convert sam to bam, sort bam and index bam:

```
$ samtools view -bt ref/hg19.fa.fai results/TCGA-AA-A01T.tumor.sam.gz > results/TCGA-AA-A01T.tumor.bam
$ samtools sort results/TCGA-AA-A01T.tumor.bam > results/TCGA-AA-A01T.tumor.sorted.bam
$ samtools index results/TCGA-AA-A01T.tumor.sorted.bam
```

The final sorted and indexed bam file should be the same as the one in /data/3.mutation folder.

2.a.5. Take a look at the alignment:

```
$ samtools view results/TCGA-AA-A01T.tumor.sorted.bam|less
```

Exercise 3: Alignment QC. What proportion of reads can be mapped to the reference genome?

2.a.6. Parse basic alignment statistics into a text file with samtools:

```
$ samtools stats results/TCGA-AA-A01T.tumor.sorted.bam > results/TCGA-AA-A01T.tumor.sorted.bam.stats
```

2.b. Advanced practice: aligner selection (novoalign), reference genome selection (with decoy sequences), and fine processing of bam files (Picard, GATK).

3. Mutation calling

3.a. Basic practice: standard pipeline (Varscan) and checking the mutation call in IGV.

Exercise 4: Call mutations with VarScan. Input data can be found in /data/3.mutation.

3.a.1. Mutation calling.

VarScan v.2.4.3 is provided under /tools. Run the following commands to call somatic mutations:

```
$ samtools mpileup -f ref/hg19.fa data/3.mutation/TCGA-AA-A01T.normal.sorted.bam data/3.mutation/TCGA-AA-A01T.tumor.sorted.bam > results/TCGA-AA-A01T.mpileup
$ java -jar tools/VarScan.v2.4.3.jar somatic results/TCGA-AA-A01T.mpileup results/TCGA-AA-A01T --mpileup 1 --output-vcf 1
$ java -jar tools/VarScan.v2.4.3.jar processSomatic results/TCGA-AA-A01T.snp.vcf
```

```
$ java -jar tools/VarScan.v2.4.3.jar somaticFilter
results/TCGA-AA-A01T.snp.Somatic.hc.vcf --output-file
results/TCGA-AA-A01T.snp.Somatic.hc.filtered.vcf
```

How many high quality somatic mutations did you find for this patient? (The final vcf file should be the same as the one in /data/4.annotation.)

Exercise 5: Visualize the mutations in IGV.

3.a.2. Launch IGV.

Mac/Linux users,

```
$ ./tools/IGV_2.3.97/igv.sh
```

PC users, in windows file explore rather than Cygwin terminal, double click igv.bat in the IGV_2.3.97 folder.

3.a.3. Load both tumor and normal bam files, type in the coordinate of one of the mutations called in the previous exercise in the search box on top as chr1:2345678. Does the somatic mutation look convincing?

3.b. Advanced practice: mutation caller selection.

4. Variant annotation

4.a. Basic practice: protein coding mutations (vcf2maf).

Exercise 6: Generate MAF (mutation annotation format) file. Input data can be found in /data/4.annotation. What genes are mutated? What genes might have altered function?

4.a.1. Run vcf2maf.pl:

```
$ perl tools/vcf2maf-master/vcf2maf.pl --input-vcf
data/4.annotation/TCGA-AA-A01T.snp.Somatic.hc.filtered.vcf --
output-maf results/TCGA-AA-A01T.somatic.snp.maf --vep-path
tools/ensembl-tools-release-
88/scripts/variant_effect_predictor/ --vep-data ref/ --
buffer-size 1000 --ref-fasta
ref/homo_sapiens/88_GRCh37/Homo_sapiens.GRCh37.75.dna.primary
_assembly.fa --cache-version 88 --filter-vcf
ref/homo_sapiens/88_GRCh37/ExAC_nonTCGA.r0.3.1.sites.vep.vcf.
gz
```

4.b. Advanced practice: non-coding mutations.

5. Results aggregation

5.a. Basic practice: summary and frequently mutated genes.

Exercise 7: Explore the text file /data/5.aggregation/CRC.somatic.SNVs.indels.maf. The relevant fields are:

Column 2: chromosome
Column 3: position
Column 7: patient id
Column 9: gene name
Column 13: predicted variant function
Column 14: variant type.

Use your favorite tool to answer the following questions:

How many patients there are? How many point mutations, insertions and deletions there are in total? Which patient has the most point mutations?

Hint, the following Unix command can give number of patients:

```
$ cat data/5.aggregation/CRC.somatic.SNVs.indels.maf |cut -
f7|sort|uniq|wc
```

Exercise 8: Answer the following questions using the data file /data/5.aggregation/CRC.somatic.SNVs.indels.maf:

How many genes are mutated (non-silent mutation)? What are the top three genes that are mutated in most number of patients? Hint, one patient may carry multiple mutations in the same gene. They should be counted as 1 at patient level.

5.b. Advanced practice: significantly mutated genes (MutSig), visualization.

6. Clinical relevance

6.a. Basic practice: actionable mutations.

Exercise 9: Answer the following questions using the data file /data/5.aggregation/CRC.somatic.SNVs.indels.maf:

9a. One of the available target therapies for colorectal cancer patients is anti-EGFR treatment. This treatment requires patients to have *EGFR* expressed and wild type *KRAS*, *NRAS* and *BRAF*. Assuming we have *EGFR* expression data, can you find out which patients are not suitable for anti-EGFR treatment based on mutation data?

9b. There are inhibitors available for BRAF. The most frequent *BRAF* mutation is V600E. How many patients carry *BRAF* V600E mutation? You need to find out the genomic coordinate of V600E mutation. These patients may benefit from combined therapy (anti-EGFR + anti-BRAF).

Exercise 10: Colorectal cancer patients characterized as micro-satellite instable (MSI) can respond to immunotherapy. Are there any patients in our cohort likely to respond to immunotherapy?

6.b. Advanced practice: natural language processing, machine learning (IBM Watson).