Somatic and Germline variant identification in Tumor and Normal sample using linux terminal

Suchitra Thapa

Team: Rosalind

Objective:

To study the provided tutorial on variant identification and reproduce it by using linux terminal.

Background:

Tumor cells are variant of the normal cell and therefore comparative genomic analysis of these two cells can provide a detail spectrum of mutation in these cells. Since there are different types of mutation, selection of sample is crucial in the comparison. Normally, for germline mutation (inherited), comparison of tumor cell with a reference genome is enough but in case of somatic mutation (after birth) , comparison of tumor cell with the same person normal cell is mandatory. Therefore, this tutorial is identifying both the somatic and germline variants in the exome sequence of normal and tumor cell from the same person.

Workflow:

The script is written in bash language and can be run in any linux terminal. The script can be found here

- *Login to the server and folder creation*
  Login to the server was done using the given username and password. If server is not available any terminal in local PC can be used. A folder was created for downloading data set and ref genome files.

- *Data Acquisition:*
  The dataset was paired end data from Normal and Tumor tissue. All four sequence file was uploaded using **wget** command via the link . The reference genome hg19 was also downloaded in similar manner but unzipped using **gunzip** command since it is a compressed file. Further a text file (list.txt) was created using **cat** command with the dataset filename (SLGFSK-N_231335 and SLGFSK-T_231336) for downstream analysis.

- *Quality control and reads mapping*

The read quality was determined by using **FastQC** tools and **MultiQC** for combined quality report. Since the dataset seems quite of a good quality but had some adapters. **Trimmomatic** tool was used to trim out the adapters. Further **FastQC** and **MultiQC** were run for the trimmed dataset to confirm the final quality. The trimmed reads were proceeded to alignment with the reference genome using **BWA MEM** tool. But before mapping the reference genome hg19 was indexed using **bwa index** command

- Post processing after mapping
  After mapping, the ouput sam file was converted to bam file then the bam file was sorted and indexed using **samtools sort** and **samtools index** command respectively. The sorted reads file was filtered with **samtools view** . The bam files were viewed using **samtoosl flagstat**. Duplicates were checked using a combination of commands i.e. **samtools collate, samtools fixmate, samtools sort** and **samtools markdup** command. After confirming the duplicates, they were removed by **samtools rmdup**. Thereafter, **bamleftalign** command was used to left-align reads around indels . Then **samtools calmd** command was used to recalibrate the read quality and finally **bamtools filter** (using <= 254 map quality parameter)was used for bam dataset to ensure high quality mapping reads before variant calling.

- Variant calling and classification
  The variants file was downloaded using **wget** command with the weblink. Then a pileup file was created using **samtools mpileup** command with the reference file and the refiltered file. The variant calling was done using the **varscan** command which created a vcf output file for each dataset. These vcf file were compressed using **bgzip** command and then indexed using **tabix** command. Further, they were merged using **bcftools merge** command.

- Creating databse for annotation
  snpEff command will be used to create a databse. For that snpEff zip file was downloaded using **wget** command and unzipped using **unzip** command.

- Variant annotation and reporting
  The called variants were annotated using snpEff command using the Homo sapiens: hg19 as a reference genome. Only functional annotation was done using SnpEff tool

Result:

Mapping result

#Results: for 335
Input Read Pairs: 10602766 Both Surviving: 10526288 (99.28%) Forward Only Surviving: 76478 (0.72%)
Reverse Only Surviving: 0 (0.00%) Dropped: 0 (0.00%)
TrimmomaticPE: Completed successfully

#Results: for 336

Input Read Pairs: 16293448 Both Surviving: 16093238 (98.77%) Forward Only Surviving: 200210 (1.23%) Reverse Only Surviving: 0 (0.00%) Dropped: 0 (0.00%)

Samtool filter result

#Result of samtools filter

samtools flagstat SLGFSK-T_231336.filtered1.bam

[W::bam_hdr_read] EOF marker is absent. The input is probably truncated
31383006 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
30349 + 0 supplementary
0 + 0 duplicates
31383006 + 0 mapped (100.00% : N/A)
31352657 + 0 paired in sequencing
15676922 + 0 read1
15675735 + 0 read2
31352657 + 0 properly paired (100.00% : N/A)
31352657 + 0 with itself and mate mapped
0 + 0 singletons (0.00% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)

samtools flagstat SLGFSK-N_231335.filtered1.bam
9053504 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
1484 + 0 supplementary
0 + 0 duplicates
9053504 + 0 mapped (100.00% : N/A)
9052020 + 0 paired in sequencing
4526163 + 0 read1
4525857 + 0 read2
9052020 + 0 properly paired (100.00% : N/A)
9052020 + 0 with itself and mate mapped
0 + 0 singletons (0.00% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5

Conclusion:
The reads were filtered, trimmed, mapped and then variants were identified.

Problem faced:
Except for some time, delay for running the tools, all the instructions could be replicated as written.