NGS Data Analysis

COMMON PROCEDURES

Prepared for you by **Genialis - Friendly Bioinformatics**
www.genialis.com | info@genialis.com

Initial Steps

1. Download your data from the sequencing center. Check the **INTEGRITY** of the downloaded sequencing files:

```
md5sum sample1.fq.gz sample2.fq.gz ... sampleN.fq.gz
```

Compare the calculated hash values to the ones provided by the sequencing center.

or

Fetch reads from SRA:

```
fastq-dump --split-files SRA_ID
```

*Requires **SRA Toolkit**: http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=software*

2. Check the **QUALITY** of the sequenced samples:

```
fastqc sample1.fq.gz sample2.fq.gz ... sampleN.fq.gz
```

*Requires **FastQC**: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/*

3. Remove the sequencing adapters and **FILTER (trim)** the reads:

```
fastq-mcf adapters.fa paired_sample1.fq
  paired_sample2.fq -o paired_sample1_filtered.fq -o
  paired_sample2_filtered.fq -q 30 -l MIN_LENGTH
```

*Requires **FastqMcf**: https://code.google.com/p/ea-utils/wiki/FastqMcf*

4. Remember to always **RE-CHECK** the quality and abundance of the filtered reads!

Mapping of the Sequencing Reads to the Reference Genome

1. Build the genome index:

```
bowtie2-build reference_genome.fa my_genome_index
```

2. Map single-end reads:

```
bowtie2 -x my_genome_index -U sample1.fq -S sample1.sam
```

or

Map paired-end reads:

```
bowtie2 -x my_genome_index -1 paired_sample1.fq -2
  paired_sample2.fq -S sample12.sam
```

*Requires **Bowtie2**: http://sourceforge.net/projects/bowtie-bio/files/bowtie2/*

Mapping of the Spliced Sequencing Reads to the Reference Genome

```
tophat2 -o "./tophat_out" --GTF file.gtf my_genome_index
  sample1.fq (sample2.fq)
```

*Requires **TopHat**: http://ccb.jhu.edu/software/tophat/index.shtml*
*Requires **Bowtie2**: http://sourceforge.net/projects/bowtie-bio/files/bowtie2/*
*Requires **Samtools**: http://sourceforge.net/projects/samtools/*

Check the Mapping Statistics

```
samtools flagstat aligned_file.bam
```

*Requires **Samtools**: http://sourceforge.net/projects/samtools/*

Variant Calling

1. Convert SAM file to BAM file and sort reads:

```
samtools view -Sb align_file.sam | samtools sort - >
  align_file_sorted.bam
```

2. Index the genome:

```
samtools faidx my_fasta_genome
```

3. Mark duplicates:

```
samtools rmdup align_file_sorted.bam dedup.bam
```

4. Convert to VCF file format:

```
samtools mpileup -g -f genome.fa dedup.bam > raw.bcf
```

5. Call SNPs:

```
bcftools view -bvcg raw.bcf > snp_candidates.bcf
```

6. Filter SNPs:

```
bcftools view snp_candidates.bcf | vcfutils.pl varFilter
-Q 20 -d 5 - > final_variants.vcf
```

*Requires **Samtools**: http://sourceforge.net/projects/samtools/*

Differential Expression Analysis

1. Assemble (novel) transcripts:

```
cufflinks -g file.gtf -b genome.fa --multi-read-correct
  aligned_file.sam/bam
```

2. Merge multiple assemblies:

```
cuffmerge -g file.gtf assembly_GTF_list.txt
```

3. Run differential expression analysis using merged GTF file produced by Cuffmerge:

```
cuffdiff -b genome.fa -N -L "case","control" -o
  "./diff_expre" --multi-read-correct file_merged.gtf
  case.bam control.bam
```

*Requires **Cufflinks**: http://cole-trapnell-lab.github.io/cufflinks/*

De-Novo Sequence Assembly (Using Velvet)

1. Prepare the input dataset (paired-end data, k-mer length 31):

```
velveth output_directory 31 -fastq.gz -shortPaired
sample1.fq.gz sample2.fq.gz
```

2. Run assembly (paired-end data, insert length 400 bp, expected coverage 14):

```
velvetg output_directory -ins_length 400 -exp_coverage 14
```

3. Choice of coverage cut-off:

```
(R) > library(plotrix)
(R) > data = reads.table("stats.txt", header=TRUE)
(R) > weighted.hist(data$short1_cov, data$lgth, breaks=0:50)
```

*Requires **Velvet**: http://www.ebi.ac.uk/~zerbino/velvet/*
*Requires **R** and **plotrix package for R**: http://www.r-project.org/*