A) Setting up the environment

1. First lets add to the locations for the tools and files used in the pipeline. For convinience all such user editable setting are in a single file by the name “base_dependencies.groovy” in the config directory. Open this file so that we can edit the tool locations and set the paths for some required files.
2. Navigate to the section starting with labelled “Changed by user”. The section should appear similar to the one shown below

```
//------------------------Changed by the user---------------------------------
//Number of threads - [REQUIRED]
NO_OF_THREADS="2"

// Reference settings The path where the required reference file is present
REFERENCE_BWA="/home/JohnDoe/References/human_g1k_v37.fa"
REFERENCE_STAMPY="/home/JohnDoe/References/References/hg19"
REFERENCE_GATK="/home/JohnDoe/References/References/human_g1k_v37.fa"
//Other input files
DBSNP_VCF_FILE="/home/JohnDoe/References/References/dbsnp132_20101103.vcf"

//--------------------Tool locations-------------------------------
//Please provide complete paths
FASTQC_LOCATION="/home/JohnDoe/Tools/FastQC/fastqc"
TRIMMOMATIC_LOCATION="/home/JohnDoe/Tools/Trimmomatic/trimmomatic-0.32.jar"
BWA_LOCATION="bwa"
SAMTOOLS_LOCATION="samtools"
STAMPY_LOCATION="/home/JohnDoe/Tools/stampy.py"
PICARD_SORTSAM_LOCATION="/home/JohnDoe/Tools/SortSam.jar"
PICARD_MARKDUP_LOCATION="/home/JohnDoe/Tools/MarkDuplicates.jar"
GATK_LOCATION="/home/JohnDoe/Tools/GenomeAnalysisTK.jar"
BAMSTATS_LOCATION="/home/JohnDoe/Tools/BAMStats-1.25.jar"
JAVA_LOCATION="java"
JAVA_MAX_MEM="28"
```

3. Edit this section and add the required details as follows:-

NO_OF_THREADS – The number of threads to be run depending on your PC configration

REFERENCE_* - The reference file location for the step

DBSNP_VCF_File – The location of dbsnp vcf file required by base recalibration step of GATK Base recalibration

In the tool locations part add the locations of the specified tools. Please remember to add only the location and not prefixes/suffixes required to run it. Also be careful not the remove the dobule quotes at the start/end of the path.

Please observe that the reference for stampy just has the prefix for the stampy indexes.

4. Also now add the location of the TRIMMOMATIC_ILLUMINA_ADAPTER_FILE in the trimmomatic secton below

5. Now moving on actually runnig the pipeline. To start with you will have to configure the flow and the tools you actually want to run. You can find this configration in the "Flow.groovy" file in the config directory. If you navigate to the bottom of this file you can see the Bpipe.run section as shown below

```
Bpipe.run {
//validation + fastqc_initial + trimmomatic + fastqc_post_trimmomatic + bwa + stampy +
validation + fastqc_initial + trimmomatic + fastqc_post_trimmomatic + bwa + complete
//validation + fastqc_initial + complete
//validation + complete
//validation + statistics_depth_of_coverage + statistics_bamstats + complete
}
```

Here you can set which tools you want to run and what should be their flow. We have included a few trial flows. To comment a flow simple add // to its start and remove it to set the flow. The names of the sections are as follows

1. validation
2. fastqc_initial
3. trimmomatic
4. fastqc_post_trimmomatic
5. bwa
6. stampy
7. picard_sortsam
8. picard_dupmark
9. gatk_indel_realign
10. gatk_base_recalibration 5
11. statistics_samtools
12. statistics_depth_of_coverage
13. statistics_bamstats
14. complete

Please have the first step as validation and the last one as complete you can edit the flow to keep onlt the rquired tools in the middle.

B) Running the pipeline

1. Once you have set the tool locations and the flow you can easily launch the pipeline by using the *pipeline_executor.pl* script. The script takes the following arguments

USAGE:
-i Directory containing the input fastq files
-o Directory to store the results. In not existing it will be created
-m Execute in multi node environment give Resource manager as argument(Eg -m lsf/ -m sge)
-j Jobname
Move all your input fastq files into a directory. The paired files should be with _1.fastq and _2.fastq naming patterns. An example dataset had been included in the test_inputs.
An example commnad to run the pipleline on the included sample dataset is

perl pipeline_executor.pl -i ./test_inputs -o ./test_run -j Test

The same command if to be executed in a multi node environment with lsf as the resource manager would be

perl pipeline_executor.pl -i ./test_inputs -o ./test_run -j Test -m lsf

C) **Viewing the Results**

The results will be created in the specific output directory. Use the sample_wise.txt and stage_wise.txt file to monitor the runs by sample/stage. The StatsandQC dir will have all the statistics and fastqc images. While the sample name wise directories will have the results of all the stages for that sample