**DNA-seq workflow (GENOME\_GPS v1.2)**

**BIC CONTACTS :**

**Ross, Christian** [Ross.Christian@mayo.edu](mailto:Ross.Christian@mayo.edu)

**Baheti, Saurabh** [Baheti.Saurabh@mayo.edu](mailto:Baheti.Saurabh@mayo.edu)

**Hart, Steven N., Ph.D.** [Hart.Steven@mayo.edu](mailto:Hart.Steven@mayo.edu)

**TEST DATASETS**

Input fastq file:

/data2/bsi/tertiary/Kocher\_Jean-Pierre\_m026645/s110943.Synthetic\_Genomes/dna-seq/fastq/

Input bam file:

/data2/bsi/tertiary/Kocher\_Jean-Pierre\_m026645/s110943.Synthetic\_Genomes/dna-seq/bam/

Input vcf/text file:

/data2/bsi/tertiary/Kocher\_Jean-Pierre\_m026645/s110943.Synthetic\_Genomes/dna-seq/vcf/

**WORKFLOW SCRIPTS**

/projects/bsi/bictools/scripts/dnaseq/GENOME\_GPS/trunk/

**EXAMPLE CONFIGURATION FILES**

/projects/bsi/bictools/scripts/dnaseq/GENOME\_GPS/trunk/config/

**README FILES**

/projects/bsi/bictools/scripts/dnaseq/GENOME\_GPS/trunk/readme/

**INTRODUCTION**

The DNA-seq Workflow (GENOME\_GPS v1.2) currently supports analysis for DNA sequencing data from multiple protocols:

1. Exome sequencing (Agilent/Nimblegen Capture Kit)

To selectively sequence the coding regions of the genome

1. Custom Capture sequencing

To selectively sequence the region of interest depending on the study

1. Whole Genome sequencing

DNA samples are extracted using the whole genome data.

1. Paired DNA sequencing

Tumor-Normal (somatic) Calling on DNA seq samples

**GENOME\_GPS v1.2** can be run using any of the following six modules depending on the data type:

The six modules are:

1. Alignment
   * Used for aligning reads against the reference genome
   * Not to be used in conjunction with paired analysis
2. mayo/external
   * Used to align the data and do variant calling with Realignment and Recalibration
   * Annotated reports are generated per sample.
3. realign-mayo/realignment
   * Annotated reports are generated per sample.
4. Variant
   * Annotated reports are generated per sample.
5. ontarget
   * Used for annotating raw variants which are in the required target region
   * Not to be used in conjunction with paired analysis
6. Annotation
   * Used for annotating list of high quality variants
   * Not to be used in conjunction with paired analysis

**NOTE:** All the modules are described in the later part of this document.

**PRE-WORKFLOW SUBMISSION / PRE-RUN**

Initial preparation:

1. Check if the FASTQC results are available.
2. Check NGS portal for consistency of samples and sample names.
3. For Exome Samples: user needs to get the capture kit information.
4. For Paired Samples: user needs to get the proper grouping information.

**PREPARING CONFIGURATION FILES**

User needs to create four configuration files for running the workflow.

1. Run Information File
2. Sample Information File
3. Tool Information File
4. Memory Information File

**Example configuration files are available at:** /projects/bsi/bictools/scripts/dnaseq/GENOME\_GPS/trunk/config/

**NOTE:** Since the configuration files at the above path are the core information files for the workflow, DO NOT make any changes to these files. For editing them, first copy them over to your working directory and then proceed in making changes relevant to your project.

**Sample Information File:**

This file contains names of the samples, names of the sample files and sample identifier.

For **external/mayo module**, since the input files will be FASTQ, we will create a sample information file as follows:

FASTQ:sampleA=sampleA.input.R1.fastq.gz sampleA.input.R2.fastq.gz

FASTQ:sampleB=sampleB.input.R1.fastq.gz sampleB.input.R2.fastq.gz

Note the “FASTQ” tag at the beginning. Also, note that in case of paired end samples, a tab separates the names of the two ends. If there are multiple fastq’s then each pair is separated by tab.

For the **realignment/realign-mayo/variant** module, the sample information file will look as follows :

BAM:sampleA=sampleA.input.bam

BAM:sampleB=sampleB.input.bam

Note the “BAM” tag at the beginning. Again, if multiple BAM files for a sample, we will separate their names with a tab.

For the annotation/ontarget module, there would be prefix tags and the sample information file would look simply as below:

SNV:sampleA=sampleA.snv.vcf

INDEL:sampleA=sampleA.indel.vcf

SNV:sampleB=sampleB.snv.vcf

INDEL:sampleB=sampleB.vcf

Note the “SNV/INDEL” tag at the beginning. We support one VCF file per sample. If users VCF file has indels and snvs both then same file should be provided for both SNV and INDEL.

**Run Information File:**

The run information file is the file that is submitted to the workflow. It contains various parameters such as paths to sample information file, tool information file, memory information file, sample names and so on.

Prepare this file according to your requirements for running this workflow. A description of the parameters in this file is as provided below:

1. TOOL: This is the name of the workflow and should not be changed and should always be GENOME\_GPS.

(Ex : TOOL=GENOME\_GPS)

1. VERSION: This is the version of the tool and should be 1.2. Please do not change this.

(Ex. VERSION=1.2)

1. TYPE: This is the type of the workflow. Please do not change this other than these two options.

(Ex. TYPE=exome/whole\_genome)

1. PAIRED: This is a flag for specifying if your reads are single-end or paired-end. ‘0’ for single-end and ‘1’ for paired-end.

(Ex. PAIRED=1)

1. READLENGTH: This is the length of your reads obtained from sequencing.

(Ex. READLENGTH=50)

1. DISEASE: If disease information is available, you can provide that here. If not, you can use “NA”.

(Ex. DISEASE=cancer)

1. PI: This is the name of the investigator in the format “LastName\_FirstName\_LANID”

(Ex. PI= baheti\_saurabh\_m078940)

1. MULTISAMPLE: This is a YES/NO flag for specifying whether you want to do paired analysis or single sample analysis.
2. INPUT\_DIR: This is the input directory where your files are located. For the “external/mayo “module, this would be the path to the directory containing FASTQ or FASTQ.GZ files. For the “realign-mayo/realignment/variant” module, this would the path to the input directory containing BAM files. For the “annotation/ontarget” module, this would be the path to the “TXT” files. All the input files should be in the directory.

(Ex. INPUT\_DIR= /data2/bsi/tertiary/Kocher\_Jean-Pierre\_m026645/s110942.methylation/pipeline/fastq/)

1. BASE\_OUTPUT\_DIR: This is the path to the base output directory, which would contain directories with the PI names and Flow cell ID’s. Most of the time, this path would be /data2/bsi/secondary/

(Ex. BASE\_OUTPUT\_DIR=/data2/bsi/secondary)

1. SAMPLENAMES: This is a colon-separated list of sample names. The naming and order of samples must match the naming from the sample information file.

(Ex. SAMPLENAMES=sampleA:sampleB)

1. GROUPNAMES=this is a colon-separated list of groups specified in the sample info file. It must match the names in the sample information file.

(Ex. GROUPNAMES=pair\_sampleA)

1. ALIGNER=the workflow gives the user to pick an aligner from two aligners i.e. bwa or novo align. Default is novo align

(Ex. ALIGNER=novoalign)

1. LANEINDEX: This is a colon-separated list of lanes on the flow cell. These must match the order of the samples from “SAMPLENAMES”. If lane information is not available, use “-”.

(Ex. LANEINDEX=1:2)

1. LABINDEXES: This is a colon-separated list of barcode tags that were used for the samples (especially if the samples were multiplexed). These must match the order of the “SAMPLENAMES”. If this information is not available, use “-“. We should ignore the “I” in the index, as this is only 6-letter code.

(Ex: LABINDEXES=AGGCCTA:CTGGACA)

1. CHRINDEX: This is a colon-separated list of chromosomes you wish to analyze.

(Ex. CHRINDEX=1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:X:Y:M)

1. VARIANT\_TYPE= this is an option to specify if user wants to call snvs or indels or both. The options are BOTH/SNV/INDEL. Default is BOTH

(Ex. VARIANT\_TYPE=BOTH)

1. SNV\_CALLER= There are three callers which user can use GATK, SNVMIX or BEAUTY\_EXOME (consensus calling)

(Ex. SNV\_CALLER=GATK)

1. SOMATIC\_CALLER= If you are doing Paired analysis then user needs to specify the caller from one of SOMATICSNIPER, JOINTSNVMIX, MUTECT or BEAUTY\_EXOME (consensus calling)

(Ex. SOMATIC\_CALLER= SOMATICSNIPER)

1. TOOL\_INFO: This is the path to the tool information file.

(Ex: TOOL\_INFO=/path/to/tool\_info.txt)

1. SAMPLE\_INFO: This is the path to the sample information file.

(Ex. SAMPLE\_INFO=/path/to/sample\_info.txt)

1. MEMORY\_INFO: This is the path to the memory information file.

(Ex. MEMORY\_INFO =/path/to/memory\_info.txt)

1. ANALYSIS: This is where the analysis type can be specified. Options are “external”, “mayo”, “realignment”, “realign-mayo”, “variant”, “ontarget” or “annotation”.

(Ex. ANALYSIS=mayo)

1. OUTPUT\_FOLDER: This is the name of the output directory that will contain the results

(Ex. OUTPUT\_FOLDER=test\_run)

1. GENOMEBUILD: This is the build of the genome used.

(Ex. GENOMEBUILD=hg19)

1. SAMPLEINFORMATION: If sample information is available, it can be specified here. This parameter is free text.

(Ex. SAMPLEINFORMATION=Tumor samples for testing)

1. FASTQC: This is a YES/NO flag for specifying whether you want to do FASTQC. Recommended option is YES if the input files are FASTQ.

(Ex. FASTQC=YES)

1. FOLDER\_FASTQC: This is the path to the folder where you have the input files for FASTQC. Since we would want to carry out FASTQC for all of our input files, this would be “NA” most of the times.

(Ex. FOLDER\_FASTQC=NA)

1. DELIVERY\_FOLDER: This is the path to the investigator’s directory where you want to deliver the results. This would be /data2/delivery/PI\_name/FlowcellID most of the times.

(Ex. DELIVERY\_FOLDER=/data2/delivery/ Jen\_Jin\_m066108/1200086\_FC10001\_02334/

1. TERTIARY\_FOLDER: This is the path to the directory, which will have results to the tertiary analysis. Use “NA” if not available. This would be /data2/bsi/tertiary/PI/type/flowcell id

(Ex. TERTIARY\_FOLDER=NA)

**Tool Information File:**

The tool information file contains paths to references and tools and you would usually not be required to change anything. However, depending upon the protocol from which you get your input data and the input data itself, you would need to change the following parameters.

HTTP\_SERVER=bmidev2

* The mayo server used to host the results for delivery to PIs

THREADS=4

* Number of threads to use during the analysis.

REORDERSAM=NO

* ReorderSam reorders reads in a SAM/BAM file to match the contig ordering in a provided reference file, as determined by exact name matching of contigs.
* Reads mapped to contigs absent in the new reference are dropped.
* YES/NO option can be used

EMIT\_ALL\_SITES=NO

* Produces calls at any callable site regardless of confidence
* YES/NO option can be used.

VARIANT\_FILTER=YES

* Flag to filter germ line calls
* YES/NO option can be used.

SOMATIC\_VARIANT\_FILTER=YES

* Flag to filter somatic calls
* YES/NO option can be used.

DEPTH\_FILTER=0

* Filter the variant calls using depth as a filter
* Should be a number

TARGETTED=YES

* Flag to only do calling on target region specified in the tool info file
* YES/NO option can be used.
* YES for Exome and custom capture samples
* NO for whole genome samples

MARKDUP=YES

* Flag to run PICARD remove duplicate
* YES/NO option can be used.

REMOVE\_DUP=FALSE

* To Remove the Duplicate reads or just flag it.
* TRUE/FALSE option can be used.

REMOVE\_ALIGNED\_BAM=NO

* To remove the aligned BAMs within the execution of the workflow
* YES/NO option can be used.

T\_DEPTH\_FILTER=6

* Filter the variant calls using depth as a filter for multi sample vcf
* Should be a number

UPLOAD\_TABLEBROWSER=YES

* To upload the variants to the Table Browser
* YES/NO option can be used.

PLATFORM=illumina

* Information about the sequencing platform

CENTER=mayo

* Information about the sequencing center

QUEUE=ngs-sec –l medp=TRUE

* Queue to use on RCF to run the workflow

#Tool Parameters

**If user doesn’t specify any thing then tool will run in the default settings**

**NOTE:** we are describing important parameters but there can be other parameters that use can use depending on the data and study.

SNVMIX2\_params=

* user can specify tool parameters in the same line separated by space

SNVMIX2\_Filter=-p 0.8

* -p filter the variants less than 0.8 probability

UnifiedGenotyper\_params=-maxAlleles 5

* User can specify tool parameters in the same line separated by space
* -maxAlleles Maximum number of alternate alleles to genotype

SOMATIC\_INDEL\_params=--window\_size 1000

* User can specify tool parameters in the same line separated by space
* --window\_size Size (bp) of the sliding window used for accumulating the coverage

SOMATIC\_SNIPER\_params=-q 20 -Q 20

* User can specify tool parameters in the same line separated by space
* -q filtering reads with mapping quality less than INT [0]
* -Q filtering somatic snv output with somatic quality less than INT [15]

MUTECT\_params=

* User can specify tool parameters in the same line separated by space

BREAKDANCER\_params=-c 5 -r 10

* User can specify tool parameters in the same line separated by space
* -c
* -r

CREST\_params=

* User can specify tool parameters in the same line separated by space

JSM\_Filter=-prob 0.1

* User can specify tool parameters in the same line separated by space
* -prob filter the somatic calls less than 0.1 probability

JOINTSNVMIX\_params=

* User can specify tool parameters in the same line separated by space

NOVO\_params=-g 60 -x 2 -i PE 425,80 -r Random --hdrhd off -v 120 -c 4

* User can specify tool parameters in the same line separated by space
* -g Sets the gap opening penalty. Default 40
* -x Sets the gap extend penalty. Default 6
* -i Sets approximate fragment length and standard deviation (comma separator) or

a range of fragment lengths (hyphen separator).

* -r Sets strategy for reporting repeats. 'None', 'Random', 'All', 'Exhaustive',
* --hdrhd off Controls checking of identity between headers in paired end reads.
* -v Sets the structural variation penalty for chimeric fragments.
* -c Sets maximum number of threads to use. Defaults to one thread per CPU Core.

BWA\_params=-l 32 -t 4

* User can specify tool parameters in the same line separated by space
* -l seed length [32]
* -t number of threads [1]

VQSR\_params\_SNV=-an QD -an HaplotypeScore -an MQRankSum -an ReadPosRankSum -an FS -an MQ -an DP -nt 2 --maxGaussians 4 --percentBadVariants 0.05

* User can specify tool parameters in the same line separated by space
* -an The names of the annotations which should used for calculations
* -nt number of threads
* --maxGaussians The maximum number of Gaussians to try during Bayes algorithm
* --percentBadVariants what percentage of the worst scoring variants to use when building the Gaussian mixture model of bad variants.

VQSR\_params\_INDEL=-an QD -an FS -an HaplotypeScore -an ReadPosRankSum --maxGaussians 4 -nt 2 --percentBadVariants 0.12 -std 10.0

* User can specify tool parameters in the same line separated by space
* -nt number of threads
* -an The names of the annotations which should used for calculations
* --percentBadVariants What percentage of the worst scoring variants to use when building the Gaussian mixture model of bad variants.
* -std If a variant has annotations more than -std standard deviations away from mean then don't use it for building the Gaussian mixture model.

PICARD\_ReadGroup\_params=PL=illumina CN=mayo LB=hg19 CREATE\_INDEX=true

* User can specify tool parameters in the same line separated by space
* PL platform information
* CN center information
* LB library information
* CREATE\_INDEX

VCF\_annotation\_params=-A QualByDepth -A MappingQualityRankSumTest -A ReadPosRankSumTest -A HaplotypeScore -A DepthOfCoverage -A MappingQualityZero -A DepthPerAlleleBySample -A RMSMappingQuality -A FisherStrand -A ForwardReverseAlleleCounts

* User can specify tool parameters in the same line separated by space
* -A One or more specific annotations to apply to variant calls

REALIGN\_params=--maxReadsForRealignment 20000 --maxReadsInMemory 150000

* User can specify tool parameters in the same line separated by space
* --maxReadsForRealignment
* --maxReadsInMemory

BLAT\_params=-w 50 -m 70 -t 90

* User can specify tool parameters in the same line separated by space
* -w window size
* -m This is twice the matches minus the mismatches minus some sort of gap penalty
* -t minIdentity

### paramters

CNVNATOR\_BINSIZE=1000

* bin size for cnv nator

PCT\_READS\_SEGSEQ=0.05

* Filters for segseq

MINFOLD=0.5

* Filters for cnv nator

MAXFOLD=1.5

* Filters for cnv nator

DISTGAP=1000

* If the SV is near to this distance from a gap then filter it.

BLAT\_PORT=50000

* port to run the blat gfserver for CREST

BLAT\_SERVER=localhost

* server to run blat gfserver for CREST

STRUCT\_DIST\_GENE=1000

* sv

STRUCT\_MIN\_SUPPORT=10

* Min #of reads required to keep the SV

STRUCT\_MIN\_IDENTITY=0.9

* Min identity to keep SV

STRUCT\_PCT\_BLACKLIST=1

* Percent

SNP\_DISTANCE\_INDEL=10

* Flag the SNV call if it is near to an INDEL

MAX\_FILE\_HANDLES=100

* number of temporary files to generate during remove duplicate from PICARD
* ulimit –n : command to check the limit of temporary files your system can handle.

Most of RCF nodes:

ulimit -n

1024

MAX\_READS\_MEM\_SORT=2000000

* Number of reads to keep in memory for sorting
* Increasing this number will use more memory.

TB\_PORT=8886

* Port for table browser

TB\_HOST=charlotte

* Host for Table Browser

JOB\_LIMIT=3000

* Number of jobs a user can submit on RCF

After the configuration files are ready, user should be ready to run the workflow.

**ANALYSIS STEPS AND SCRIPT CALLS**

The following section details the steps that each module takes for analysis. It also describes which scripts are called. “🡪” and “🡻”denotes script calls.

**Alignment Module:**

This Module allows user to align the data against the reference genome using the aligners available and mentioned in the configuration files. There are two options of the aligner in this workflow i.e. bwa or novoalign. (default is novoalign)

**INPUT:** Single-end/Paired-end FASTQ or FASTQ.GZ

**OUTPUT:**

* Sorted and indexed BAM file
* MainDocument.html
* SampleStatistics.tsv

**NOTE:**

1. This module will stop after the alignment is completed and will generate a report and statistics for the alignment.
2. This option is only available for exome/custom capture and whole genome data.
3. This module should not be used in conjunction with paired analysis.

The analysis steps are similar for all protocols with minor differences which will be detailed as and when relevant.

The **alignment** module proceeds through the following steps of analysis for every input sample:

1. The input FASTQ file is first run through FASTQC to generate QC reports.

(whole\_genome.sh 🡪 align\_novo.sh/align\_read\_bwa.sh 🡪 fastq.sh)

1. After QC, the reads are aligned using the aligner mentioned in the run info and sorted aligned BAM file is generated

(whole\_genome.sh 🡪 align\_novo.sh/align\_bwa.sh 🡪 convert\_bam.sh)

1. If there are multiple pairs of fastq’s are available then all will run in parallel and then merged back to generate single sorted BAM file with a statistics about the BAM. The BAM file will be sorted and read group will be added to the BAM file if it doesn’t exist before. Duplicates reads are also flagged in the BAM file using PICRAD

(whole\_genome.sh 🡪 processBAM.sh 🡪 PICARD & SAMTOOLS)

1. Then statistics is computed using Unix and shell tools and Samtools flag stat.

**Scripts Flow:**

whole\_genome.sh

🡻

align\_novo.sh/align\_bwa.sh 🡪 fastq.sh🡪 convert\_bam.sh 🡪 sortbam.sh 🡪 addReadGroup.sh

🡻

processBAM.sh 🡪 MergeBam.sh 🡪 sortbam.sh 🡪 addReadGroup.sh 🡪 rmdup.sh

🡻

sample\_numbers.sh

🡻

generate\_html.sh

**Scripts used**:

checkBAMsorted.pl

checkFastqQualityScores.pl

checkBAMreadGroup.pl

**Mayo/External Module:**

The Mayo and the External modules are essentially the same, the only difference being the Mayo module updates the secondary dashboard while the external module does not.

**INPUT:** Single-end/Paired-end FASTQ or FASTQ.GZ

**OUTPUT:**

* Sorted and indexed IGV bam file
* Annotated variant Reports
* Merged variant reports for all the samples
* IGV session files
* MainDocument.html
* SampleStatistics.tsv

The analysis steps are similar for all three protocols with minor differences, which will be detailed as and when relevant.

The **mayo/external** module proceeds through the following steps of analysis for every input sample:

1. The input FASTQ file is first run through FASTQC to generate QC reports.

(whole\_genome.sh 🡪 align\_novo.sh/align\_read\_bwa.sh 🡪 fastq.sh)

1. After QC, the reads are aligned using the aligner mentioned in the run info and sorted aligned BAM file is generated

(whole\_genome.sh 🡪 align\_novo.sh/align\_bwa.sh 🡪 convert\_bam.sh)

1. If there are multiple pairs of fastq’s are available then all will run in parallel and then merged back to generate single sorted BAM file with a statistics about the BAM. The BAM file will be sorted and read group will be added to the BAM file if it doesn’t exist before. Duplicates reads are also flagged in the BAM file using PICRAD

(whole\_genome.sh 🡪 processBAM.sh 🡪 PICARD & SAMTOOLS)

1. These bam files are then re-aligned and re-calibrated for every input chromosome defined in the run information file.

(whole\_genome.sh 🡪 realign\_reacal.sh🡪 realign\_per\_chr.sh 🡪recal\_per\_chr.sh)

1. Optionally, if the data type is whole genome then structural varaints tools are executed using the realigned and aligned bam file

(whole\_genome.sh 🡪 run\_single\_crest.sh)

(whole\_genome.sh 🡪 run\_breakdancer.sh)

(whole\_genome.sh 🡪 run\_cnvnator.sh )

1. If step 5 is executed then per sample VCF files for structural varaints are merged and stored in reports folder.

(whole\_genome.sh 🡪summaryze\_struct\_single.sh)

1. These realigned bam files are used for variant calling using the variant caller mentioned in the run information file

(whole\_genome.sh 🡪 varaints.sh)

1. The VCF files obtained from the callers are then merged for each sample and filtered using VQSR from GATK. If VQSR fails to apply then hard filters are applied to the data.

(whole\_genome.sh 🡪 merge\_variant\_single.sh 🡪 concatvcf.sh 🡪 filter\_variant\_vqsr.sh 🡪 hardfilters\_variants.sh)

1. The filtered variants are then subset to the target region specified in the tool information file for Exome and capture data and for whole genome data it is subseted to the gene body

(whole\_genome.sh 🡪 OnTarget\_variant.sh)

1. In parallel to the variant calling the coverage and ontarget statistics are also computed on each bam file generated after the realignment and recalibration.

(whole\_genome.sh 🡪OnTarget\_BAM.sh)

(whole\_genome.sh 🡪 OnTarget\_PILEUP.sh)

1. After the VCF files are intersected with the target region the annotation module start which includes annotations from SIFT, SNPEFF and POLYPHEN

(whole\_genome.sh 🡪 sift.sh)

(whole\_genome.sh 🡪 snpeff.sh)

(whole\_genome.sh 🡪 polyphen.sh)

1. All the annotation are merged and summarized in the excel worksheet

(whole\_genome.sh 🡪 reports.sh)

1. The annotations from all chromosomes are then merged for each sample.

(whole\_genome.sh 🡪 sample\_report.sh)

1. Statistics from bams, variant calls, capture kits and annotated reports are then generated for each sample.

(whole\_genome.sh 🡪 sample\_numbers.sh)

1. High level gene summary file is generated using the annotated reports for each sample.

(whole\_genome.sh 🡪 gene\_summary.sh)

1. Merged annotated report is generated for all the samples after the completion of all the individual samples.

(whole\_genome.sh 🡪 merge\_sample.sh )

1. A final report (Main\_Document.html) is then generated for each sample containing these statistics.

(whole\_genome.sh 🡪 generate.html.sh)

Single Sample: Exome

whole\_genome.sh

🡻

align\_novo.sh/align\_bwa.sh🡪fastq.sh🡪convert\_bam.sh🡪sortbam.sh🡪addReadGroup.sh

🡻

processBAM.sh🡪MergeBam.sh🡪 sortbam.sh🡪addReadGroup.sh🡪rmdup.sh

🡻

extract\_reads\_bam.sh🡪MergeBam.sh

🡻

realign\_recal.sh🡪realign\_per\_chr.sh🡪recal\_per\_chr.sh

🡻 🡻 🡻

igv\_bam.sh OnTarget\_BAM.sh OnTarget\_PILEUP.sh

🡻

getcoverage.sh

🡻

variants.sh🡪unifiedgenotyper.sh🡪annotate\_vcf.sh

🡻

merge\_variant\_single.sh🡪

🡻

OnTarget\_variants.sh

🡻 🡻 🡻

sift.sh snpeff.sh polyphen.sh

🡻 🡻 🡻

🡻

reports.sh

🡻

sample\_report.sh

🡻

sample\_numbers.sh

🡻

gene\_summary.sh

Two scripts are run after the completion of all the per sample scripts

Merge\_sample.sh

🡻

Generate\_html.sh

Single Sample: whole genome

whole\_genome.sh

🡻

align\_novo.sh/align\_bwa.sh🡪fastq.sh🡪convert\_bam.sh🡪sortbam.sh🡪addReadGroup.sh

🡻

processBAM.sh🡪MergeBam.sh🡪 sortbam.sh🡪addReadGroup.sh🡪rmdup.sh

🡻

extract\_reads\_bam.sh🡪MergeBam.sh

🡻

realign\_recal.sh🡪realign\_per\_chr.sh🡪recal\_per\_chr.sh

🡻 🡻 🡻 🡻 🡻

igv\_bam.sh OnTarget\_BAM.sh OnTarget\_PILEUP.sh run\_cnvnator.sh

🡻

getcoverage.sh

🡻

variants.sh🡪unifiedgenotyper.sh🡪annotate\_vcf.sh

🡻

merge\_variant\_single.sh🡪

🡻

OnTarget\_variants.sh

🡻 🡻 🡻

sift.sh snpeff.sh polyphen.sh

🡻 🡻 🡻

🡻

reports.sh

🡻

sample\_report.sh

🡻

sample\_numbers.sh

🡻

gene\_summary.sh

Two scripts are run after the completion of all the per sample scripts

Merge\_sample.sh

🡻

Generate\_html.sh

SCRIPTS description:

**./align\_novo.sh**

**wrapper script to run the alignment using NOVO ALIGN**

**Usage: ./align\_novo.sh <sample name> </path/to/output\_dir> </path/to/run\_info.txt> <SGE TASK ID (optional)>**

Purpose: The script allows user to align the fastq's (PE or SR) using novoalign as an aligner. The script checks for the Quality scores of the fastq's (Illumina or Sanger score). It also runs the fastqc for the fastq depending on the flag set in the run info configuration file.

Waits On: primary script

Modules: external, mayo or alignment

Inputs: sample name, output directory, run info configuration file

Outputs: aligned, sorted and Indexed BAM file with Alignment Statistics.

Output Files: \*.bam \*.bam.bai \*.flagstat

Warnings/Errors:

• Sorting of BAM fails: due to number of splits a system will allow. For the sorting process we use picard tool which during the sorting splits out tmp files on the disc and each disk has some limitations of number of files a tool can split out.

$ ulimit –n

(You can increase the number of reads to keep into the RAM while sorting, to decrease the number of tmp files generated. But this will increase the RAM required to the process.

Scripts Used:

checkFastqQualityScores.pl

fastq.sh

convert\_bam.sh

checkBAMsorted.pl

addReadGroup.sh

sortbam.sh

Tools/Scripting Languages Used:

Novo align

FASTQC

samtools

perl

Bash Scripting

JAVA

**./align\_read\_bwa.sh**

**wrapper script to run the alignment using BWA**

**Usage: ./align\_read\_bwa.sh <sample name> </path/to/output\_dir> <which read(1/2)> </path/to/run\_info.txt> <SGE TASK ID (optional)>**

Purpose: The script allows user to align the fastq's using BWA as a aligner. BWA is a two-step process first it align the individual read and create a \*.sai file(binary file) and then depending on the read type (PE or SR) it used the module of BWA to generate a aligned SAM file. The script checks for the Quality scores of the fastq's (Illumina or Sanger score). It also runs the fastqc for the fastq depending on the flag set in the run info configuration file.

Waits On: primary script

Inputs: sample name, output directory, read information (1 or 2), run info configuration file

Outputs: Binary aligned file for specified Read and for samples

Output Files: \*.sai

Warnings/Errors:

Scripts Used:

checkFastqQualityScores.pl

fastq.sh

Tools/Scripting Languages Used:

BWA

FASTQC

perl

Bash Scripting

JAVA

./align\_bwa.sh

wrapper script to run the alignment using BWA

Usage ./align\_bwa.sh <sample name> </path/to/output\_dir> </path/to/run\_info.txt> <SGE TASK ID (optional)

Purpose: This script takes the \*.sai file generated using the BWA aln script to create an alignment file depending on the type of the read.

Waits On: align\_read\_bwa.sh

Inputs: sample name, output directory, run info configuration file

Outputs: BWA aligned sorted and Indexed BAM file with Alignment Statistics.

Output Files: \*.bam \*.bam.bai \*.flagstat

Warnings/Errors:

Scripts Used:

Convert\_bam.sh

checkBAMsorted.pl

addReadGroup.sh

sortbam.sh

Tools/Scripting Languages Used:

BWA

Samtools

PICARD

perl

Bash Scripting

JAVA

**./convert\_bam.sh**

**wrapper to add read group and sort and reorder the bam**

**Usage: convert\_bam.sh </path/to/input directory> <name of BAM > <sample name> <sge task id> </path/to/run\_info.txt>**

Purpose: to convert and prepare the aligned BAM file by doing sorting and checking if the Read Group and Platform information is available in the data.

Waits On: it is the part of align\_novo.sh and align\_bwa.sh

Inputs: input directory, name of the BAM file, sample name

Outputs: aligned and sorted BAM file ready for downstream analysis

Output Files: \*.bam \*.bam.bai \*.flagstat

Warnings/Errors:

Scripts Used:

checkBAMsorted.pl

addReadGroup.sh

sortbam.sh

Tools/Scripting Languages Used:

Samtools

PICARD

perl

Bash Scripting

JAVA

**./processBAM.sh**

**wrapper to merge bam files and validate the bam for downstream analysis**

**Usage: ./processBAM.sh </path/to/input directory><sample name> </path/to/run\_info.txt>**

Purpose: The script allows user to merge all the alignment BAM files for same sample and then validate the BAM file is suitable for downstream tools. It chekcs if the final BAM is sorted and remove duplicates. It checks for read Group and Platform inforamtion in the BAM. It checks for the proper order of the header with reference to the reference genome used depending on the flag in the run info file (REORDER=YES/NO).

Waits On: align\_bwa.sh or align\_novo.sh (for external, mayo or alignment module) primary script for other modules.

Inputs: input directory, sample name, run info configuration file

Outputs: sorted and Indexed BAM file for a sample with Alignment and duplicate Statistics.

Output Files: \*.bam \*.bam.bai \*.flagstat \*.dup.meterics

Warnings/Errors:

Scripts Used:

checkFastqQualityScores.pl

checkBAMsorted.pl

checkBAMreadGroup.pl

addReadGroup.sh

sortbam.sh

MergeBam.sh

reorderBam.sh

Tools/Scripting Languages Used:

Samtools

PICARD

perl

Bash Scripting

JAVA

**./extract\_reads\_bam.sh**

**script to extract reads not used for downstream processing**

**Usage: ./extract\_read\_bam.sh </path/to/input diretcory><bam file></path/to/run info></path/to/igv folder><single/pair>**

Purpose: This script allows user to keep the reads in a separate bam file which are not used for realignment and variant calling

Waits On: processBAM.sh

Inputs: input directory, bam file name, run info configuration file, path to igv folder and single or pair flag

Outputs: sorted and Indexed BAM file for a sample with extra reads

Output Files: \*.bam \*.bam.bai

Warnings/Errors:

Scripts Used:

MergeBam.sh

Tools/Scripting Languages Used:

Samtools

PICARD

Bash Scripting

JAVA

**./realign\_recal.sh**

**wrapper script to do realignment and variant calling**

**Usage: ./realign\_recal.sh Multi-Samples**

**<input ':' sep> <bam ':' sep[normal:tumor1:tumor2:tumorN]> <samples ':' sep[normal:tumor1:tumor2:tumorN]> <outputdir bams> <run\_info><1 for realign-recal or 0 for recal-realign><SGE\_TASK\_ID (optional)>**

**else**

**<input> <bam > <samples> <outputdir bams><run\_info><1 for realign-recal or 0 for recal-realign><SGE\_TASK\_ID (optional)>**

Purpose: this wrapper is used for realignment and recalibration of a chromosome chopped BAM file

Waits On: processBAM.sh

Inputs: input directory, bam file name, output directory, run info configuration file, flag to mention realign should follow recalibration or other way round. (1/0)

Outputs: sorted and Indexed per chromosome BAM file with flagstat statistics

Output Files: \*.cleaned.bam \*.cleaned.bam.bai

Warnings/Errors:

Scripts Used:

realign\_per\_chr.sh

recal\_per\_chr.sh

samplecheckBAM.sh

Tools/Scripting Languages Used:

Samtools

PICARD

Bash Scripting

JAVA

**./variants.sh**

**script to run variant calling on set of BAM files**

**Usage: ./variants.sh </path/to/input dir><samples ':' sep[normal:tumor1:tumor2:tumorN]> </path/to/output directory> <1 or 0 chopped or not ><path/to/run info><SGE\_TASK\_ID(optional)>**

**NOTE: first sample is considered as normal and others are considered as tumor [Assumption]**

Purpose: this wrapper allows user to run variant calling depending on the flags in the run information file in terms of which SNV caller or somatic caller to use.

Inputs: input directory, sample names, output directory, run info configuration file, flag to chop the bam file, if it already chopped or not(1/0), run information file.

Outputs: raw vcf file and multi vcf file (single/multi sample) raw somatic vcf and multi somatic vcf (paired sample)

Output Files: <sample>.\*.raw.vcf, <sample>.\*.raw.multi.vcf (single sample)

variants.\*.raw.vcf, varaints.\*.raw.multi.vcf, MergeAllSamples.\*.raw.vcf, MergeAllSamples.\*.multi.raw.vcf

Warnings/Errors:

Scripts Used:

unifiedgenotyper.sh

somaticindel.sh

mutect.sh

snvmix2.sh

snvmix\_to\_vcf.pl

mutect2vcf.pl

convertvcf.pl

Jointsnvmix.sh

jsm2vcf.pl

annotate\_vcf.sh

concatvcf.sh

combinevcf.sh

vcf\_blat\_verify.pl

vcfsort.pl

samplecheckBAM.sh

Tools/Scripting Languages Used:

Samtools

PICARD

Bash Scripting

JAVA

GATK

MUTECT

SNVMIX2

JOINTSNVMIX

PERL

VCFTOOLS

**script to add allele frequency for hapmap and 1kgenome populations**

**Usage ./add\_3population.sh [input file with chr as 1st column and position as 2nd column, 1-based] [outout name, e.g., run70 ]**

**script to add allele frequencies to the reported variantcalls**

**Usage: ./add.frequencies.sh </path/to/TempReportDir><variant file with rsids ><chromosome></path/to/run info>**

**script to add read group and paltform information to a BAM file**

**Usage: ./addReadGroup.sh <input bam> <outputbam></path/to/temp dir></path/to/run info><sample name>**

**script to add rsids to the indel report**

**Usage: ./add.rsids\_indels.sh </path/to/TempReportDir> <indel file><chromosome> </path/to/run info>**

**Usage<TempReportDir> <snv file><chromosome> <run info>**

**Usage: wrapper script to run the alignment using NOVO ALIGN**

**align\_bwa.sh <sample name> </path/to/output\_dir> </path/to/run\_info.txt> <SGE TASK ID (optional)**

**wrapper script to run the alignment using NOVO ALIGN**

**Usage: ./align\_novo.sh <sample name> </path/to/output\_dir> </path/to/run\_info.txt> <SGE TASK ID (optional)>**

**wrapper script to run the alignment using NOVO ALIGN**

**Usage: ./align\_read\_bwa.sh <sample name> </path/to/output\_dir> </path/to/run\_info.txt> <SGE TASK ID (optional)>**

**script to annotate vcf files using GATK**

**Usage: ./annotate\_vcf.sh <input vcf file><run info file></path/to/input directroy>**

**script to create a annotated report for all teh CNV reported for a sample**

**Usage: </path/to/output dir> </path/to/run\_info.txt> <sample/group>**

**script to get annotated report for SNV reported for a sample**

**Usage: </path/to/output dir> </path/to/run\_info.txt>c < output folder> <smaple/group name>**

**script to check the number of jobs a user can submit**

**Usage: ./check\_qstat.sh <limit of jobs>**

**script to combine multiple vcf files**

**Usage: ./combinevcf.sh <input files><output vcf ><run info><to delete input files(yes/no)**

**Usage : </path/to/input folder> <sample name> </path/to/output folder> </path/to/run info file>**

**script to concat per chromosome file to a merged file**

**Usage : ./concat\_raw\_variants.sh </path/to/output folder> </path/to/run info file>**

**script to concat the vcf files uisng vcftools**

**Usage: ./concatvcf.sh <input files><output vcf ><run info><to delete input files(yes/no)**

**wrapper to add read group and sort and reorder the bam**

**Usage: convert\_bam.sh </path/to/input directory> <name of BAM > <sample name> <sge task id> </path/to/run\_info.txt>**

**Usage: wrapper to copy the config files\n <output dir ><run\_info>**

**Usage: wrapper to create folder structure\n <run info file>**

**Usage : script to update secondary dashboard**

**Usage:<sample ><runinfo ><stage of the workflow> <status of the stage> <SGE TASK ID(optional)>**

**In buit QC script to send an email if something fails**

**Usage: ./email.sh <filename> <error message><previous script><runinfo>**

**script to report error in recommnended format**

**Usage: errorlog.sh <file name><script\_name><ERROR/WARNING> <free tesxt>**

**script to extract reads not used for downstream processing**

**Usage: ./extract\_read\_bam.sh </path/to/input diretcory><bam file></path/to/run info></path/to/igv folder><single/pair>**

**SCRIPT to run fastqc on zipped or unzipped fastq**

**Usage: ./fastq.sh <input fastq file> </path/to/input dir> </path/to/output dir> </path/to/run info> </path/to/FASTQC directory>**

**script to create a file size file to be uploaded onto the database**

**Usage: ./filesize.sh <analysis type><sample name><filename><job id ><size of the file><run info >**

**Script to filter the variants using VQSR**

**Usage: ./filter\_variant\_vqsr.sh <raw vcf complete path><outputfile><type={BOTH,SNP,INDEL}><run info>**

**script to filter the vcf using the expression**

**Usage: ./filtervcf.sh <inputvcf><runinfo file>**

**Usage: to plot coverage plot**

**<input directory><output dir><run info >**

**script to generate html and send an email to the user of workflow completion**

**Usage: <output dir> <run info file>**

**Usage: </path/to/output folder><path/to/run info file>**

**script to get the gene summary file with information about the vaianats and SVs\nUsage: </path/to/output directory> <sample name> </path/to/run\_info.txt> </path/yo/Reports\_per\_Sample>**

**script to merge the coverage numbers per sample from per chromosome**

**Usage:./getCoverage.sh <input dir><output dir><sample><run info >**

**Script to appy hard filters on vcf variant files**

**Usage: ./hardfilters\_variants.sh script to appy hard filters**

**<input vcf > <output vcf ><run info file><type SNP/INDEL>**

**SCRIPT to create IGV BAM**

**Usage: ./igv\_bam.sh </path/to/realign dir> </path/to/output folder> <sample> </path/to/alignment folder></path/to/run ifno>**

**Usage:<Tempreports><sseq dir><chromsome><variant file><run info>**

**script to run joint snvmix on a set of tumor normal bam files\nUsage: ./Jointsnvmix.sh <normal bam> <tumor bam > <output dir> <chromosome> <tumor sample name> <normal sample name ><output file> <run info>**

**script to merge multiple BAm files and outputs a sorted BAM**

**Usage: ./mergeBam.sh <input bam> <outputbam><temp dir><indexing flag (true/false)><run info>**

**script to add annotations to the indel file**

**Usage: ./merge.indel.sh </path/to/TempReports> <sample> <which\_chr> </path/to/snpeff> <indel\_file></path/to/run info>**

**script to merge per chr per sample file to merge file per chr**

**Usage : ./merge\_raw\_variants.sh </path/to/output folder> </path/to/run info file>**

**script to merge the per sample report**

**Usage : <output\_dir> <run\_info>**

**script to add various annotation to the reported varaint file**

**Usage: ./merge.snv.sh </path/to/Tempreports><sample name><chromosome></path/to/sift dir></path/to/snpeff dir></path/to/polyphen dir><variant file><run info>**

**Usage: </path/to/input directory> <group name> </path/to/output directory> </path/to/run\_info.txt>**

**script to merge the varaiants and then apply filters to the vcf file**

**Usage: ./merge\_variant\_single.sh </path/to/input directory> <sample name> </path/to/output folder> </path/to/run\_info.txt>**

**script to mrege the vcf files using vcftools**

**Usage: ./mergevcf.sh <input files><output vcf ><run info><to delete input files(yes/no)**

**Usage: <Please specify path to run\_info.txt file>**

**script to run somatic caller called mutect**

**Usage: ./mutect.sh <normal bam> <tumor bam > <output dir> <chromosome> <tumor sample name> <normal sample name> <output vcf file name> <run info>**

**Usage : SCRIPT to get Ontaget reads**

**Usage: ./OnTarget\_BAM.sh </path/to/input sample realignment></path/to/output Ontarget><sample></path/to/run info><SGE\_TASK\_ID(optional)>**

**SCRIPT to get Ontarget pileup**

**Usage: ./OnTarget\_PILEUP.sh </path/to/input dir> </path/to/output Ontarget> <sample> </path/to/run ifno><SGE\_TASK\_ID(optional)>**

**script to get ontarget varaints uisng the capture kit specified**

**Usage: ./OnTarget\_variants.sh </path/to/output directory for variants> </path/to/output directory for OnTarget> <sample name> </path/to/run\_info.txt><SGE\_TASK\_ID(optional)>**

**script to parse a vcf file to tab delimited file using vctools vcf-query**

**Usage: ./parse.vcf.sh <in\_vcf\_file> <out\_file> <run\_info><type of variant>**

**script to run phaseByTransmission**

**Usage: ./phaseByTransmission.sh <vcf input> <vcf output> <run info file>**

**script to make cirocs plot for structural variant and copy number variantion per sample**

**Usage: <sv file break> <sv file crest><cnv file> <sample> <output dir> <run info>**

**script to run polyphen on a vcf file**

**Usage: ./polyphen.sh </path/to/polphen dir> <path/to/ontarget directory><sample><run info><somatic/germline><SGE\_TASK\_ID(optional)>**

**Usage:<sample> <tempReport dir> <run info> <input variant folder><chromosome>**

**wrapper to merge bam files and validate the bam for downstream analysis**

**Usage: ./processBAM.sh </path/to/input directory><sample name> </path/to/run\_info.txt>**

**Usage:**

**1. if more than one flowcell, provide full path to text file listing flowcells --> just the flowcell names. Else provide flowcell name**

**2. output directory for plot**

**3. script path**

**4. 0 or 1 value --> 0 if plot required for only one flowcell, 1 if plot required for more than one flowcell**

**\nUsage: <list of flowcell names> </path/to/delivery folder for PI>**

**Usage:**

**1. input directory containing AllSamples\_GeneCount.noreads**

**2. output directory for plot**

**3. script path**

**script to realign the bam file uisng paramters from tool info file**

**Usage:**

**If user wants to do realignment fist**

**<input dir ':' sep><input bam ':' sep><outputdir><run\_info><1 or 0 if bam is per chr><1 for realign first><sample ':' sep>**

**else**

**<input dir><input bam><output dir><run\_info> <1 or 0 if bam is per chr> < 0 for realign second><sample(add a dummy sample name as we dont care about the sample name (example:multi))>**

**wrapper script to do realignment and variant calling**

**Usage: ./realign\_recal.sh Multi-Samples**

**<input ':' sep> <bam ':' sep[normal:tumor1:tumor2:tumorN]> <samples ':' sep[normal:tumor1:tumor2:tumorN]> <outputdir bams> <run\_info><1 for realign-recal or 0 for recal-realign><SGE\_TASK\_ID (optional)>**

**else**

**<input> <bam > <samples> <outputdir bams><run\_info><1 for realign-recal or 0 for recal-realign><SGE\_TASK\_ID (optional)>**

**script to run recalibration on a bam file uisng tool info paramters**

**Usage:**

**If user wants to do recalibration fist**

**<input dir ':' sep><input bam ':' sep><outputdir><run\_info><1 or 0 if bam is per chr><1 for recalibrate first ><sample ':' sep>**

**else**

**<input dir><input bam><output dir><run\_info> <1 or 0 if bam is per chr> < 0 for recal second><sample (a dummy sampel name i would say just type multi as sample>**

**wrapper to merge bam files and validate the bam for downstream analysis**

**Usage: ./reformat\_BAM.sh </path/to/input directory> <sample name> </path/to/run\_info.txt>**

**Usage: wrapper to merge bam files and validate the bam for downstream analysis**

**Usage: ./reformat\_pairBAM.sh </path/to/input directory> <group name> </path/to/run\_info.txt>**

**script to reformat the input vcf to run the ontarget module of the worflow**

**Usage: ./reformat\_VARIANTS\_OnTarget.sh <input folder>>/path/to/output folder><sample><run\_info><marker>**

**script to reformat the vcf files for the annotation module of the workflow**

**Usage: ./reformat\_VARIANTs.sh </path/to/output><sample><run\_info><marker>**

**script to reorder the bam file**

**Usage: <input bam> <outputbam><temp dir><run info>**

**script to merge per chromosome report to a single merged report for delivery**

**Usage: ./reports.sh <run info> <sample> </path/to/Temp reports> </path/to/Output OnTarget> </path/to/sift> </path/to/snpeff> </path/to/polyphen> </path/to/output dir><somatic/germline><SGE\_TASK\_ID(optional)>**

**script to remove or flag the duplicates from a BAM file dending on the flag passed**

**Usage: ./rmdup.sh <input bam> <outputbam><temp dir><max files to split><remove of flag dupluicate(true/false)><assume file is aorted or not(true/false)><do indexing or not(true/false)<run info>**

**script to run break dancer for single or multiple samples**

**Usage: samplenames input\_bams </path/to/output directory> </path/to/run\_info.txt> <group name>**

**script to run cnvnator on a bam file**

**Usage: ./run\_cnvator.sh <samplename> <input folder> </path/to/output directory> </path/to/run\_info.txt>**

**Script to run crest on a paired sample**

**Usage: ./run\_crest\_multi\_cover.sh <sample name> <group name> </path/to/input directory> </path/to/output directory> </path/to/run\_info.txt>**

**Script to run crest on a paired sample**

**Usage: ./run\_crest\_multi.sh <group name> </path/to/input directory> </path/to/output directory> </path/to/run\_info.txt><SGE\_TASK\_ID (optional)>**

**script to run seg seq on tumor normal bam file**

**Usage: <groupname> </path/to/input directory> </path/to/output directory> </path/to/run\_info.txt>**

**script to run CREST on a bam file**

**Usage: <samplename> </path/to/align directory/> <bam file name> </path/to/output directory> </path/to/run\_info.txt><SGE\_TASK\_ID(optional)>**

**wrapper to validate bam file**

**Usage: ./samplecheckBAM.sh <input dir><input bam><outputdir><run\_info><sample><1 or 0 if bam is per chr><which \_chr>**

**script to get statistical numbers for a sample/group**

**Usage: </path/to/input dir> <sample> </path/to/run info></path/to/number folder>**

**script to merge the per chr report**

**Usage: ./sample\_report.sh </path/to/output\_dir> </path/to/TempReports> <sample name> </path/to/run\_info><somatic/germline><SGE\_TASK\_ID(optional)>**

**script to run SIFT annotation tool on a vcf file**

**Usage: ./sift.sh <sift dir> <input dir><sample><run info><somatic/germline><SGE\_TASK\_ID(optional)>**

**script to run snpeff on a vcf file in a folder**

**Usage: ./snpeff.sh <snpeff dir> <input dir><sample><run\_info><somatic/germline><SGE\_TASK\_ID(optional)>**

**script to run snvmix**

**Usage: ./snvmix2.sh <sample> <bam file><output vcf> <mode><bed file><run info>**

**script to run somtic indel caller**

**Usage: ./somaticindel.sh <normal bam ><tumor bam><chromosome><range><tumor sample><output dir><output file vcf><run info>**

**script to run somatic snipper on the tumor normal bam file and outputs a vcf file**

**Usage: ./somaticsipper.sh <normal bam> <tumor bam > <output dir> <chromosome> <tumor sample name> <normal sample name> <output vcf file name> <run info>**

**Script to sort the bam file using picard samtools**

**Usage: ./sortbam.sh <input bam> <outputbam></path/to/temp dir><sorting order><flag for indexing(true/false)></path/to/run info>**

**script to split the bam file per chromosome (assuimg the file name as <sample>.sorted.bam)**

**Usage: ./split\_bam\_chr.sh </path/to/input directory> <sample name> </path/to/run\_info.txt><SGE\_TASK\_ID(optional)>**

**SCRIPT to split the bam uisng read group information**

**Usage: ./split\_sample\_pair.sh </path/to/realign dir> </path/to/output folder> <sample> </path/to/alignment folder></path/to/run info>**

**script to merge somatic SVs for a sample**

**Usage: <group name> <base dir> <path to run\_info file>**

**script to merge the structural varaints from multiple tools**

**Usage: <sample name> <base dir> <path to run\_info file>**

**wrapper to clean intermediate files and tansfer the data to tertiary, delivery folder**

**Usage: ./transfer\_clean.sh </path/to/secondary folder> <path/to/run\_info>**

**script to run unified genotyper**

**Usage: ./unifiedgenotyper.sh <bams><vcf output><type of varint> <range of positions> <output mode><run info file>**

**script to get the unique id for execution of the workflow**

**Usage: ./unique\_id.sh </path/to/run info file>**

**script to run variant calling on set of BAM files**

**Usage: ./variants.sh </path/to/input dir><samples ':' sep[normal:tumor1:tumor2:tumorN]> </path/to/output directory> <1 or 0 chopped or not ><path/to/run info><SGE\_TASK\_ID(optional)>**

**NOTE: first sample is considered as normal and others are considered as tumor[Assumption]**

**script to wait for the fix**

**usage: ./wait.sh </path/to/filename>**

**Wrapper script to submit all the jobs for dna-seq workflow**

**Usage: ./whole\_genome.sh <Please specify full /path/to/run\_info file>**

**RUNNING THE WORKFLOW / SUBMITTING JOB TO SGE**

The workflow should always be run on the Sun Grid Engine. If you want to run this workflow on one node machine then user needs to modify the whole\_genome.sh script.

It can be run on SGE by using the following command:

**$ qsub -V -cwd -q ngs-sec -l medp=TRUE -m ae -M your\_email@mayo.edu /path/to/scripts/whole\_genome.sh /path/to/run\_info\_file**

**NOTE** : Please provide full path to run information file.

**POST WORKFLOW COMPLETION PROCESS / POST-RUN**

1. Check “errorlog” and “warninglog” files. If they are not empty, check out the errors.
2. Check Main\_Document.html and see if there are consistent statistics for all columns.
3. Check the “Reports\_per\_sample” folder to see if there are annotated reports for each sample.
4. Check the “numbers” folder for each sample.
5. Check “StatisticsDescriptions.html” to see if all columns are filled.
6. Compare the numbers from “StatisticsDescriptions.html” and “SampleStatistics.tsv” file to see if they are the same.
7. Check if “IGV\_Setup.doc” is updated.
8. Run RSync on the Isilon server.
9. Copy folders (??) to the PI\_Support\_Projects space.
10. Update Project\_Tracking.xls sheet.