TASSEL 3 UNEAK Pipeline

#Step 1 create working directories

${TASSEL3} -Xms30G -Xmx30G -fork1 -UCreatWorkingDirPlugin -w ${WORKDIR} -endPlugin -runfork1

#This creates working directores, once the key and fastq files are entered in the directories, the user does not have to specify paths to files.

#Step 2

#copy fastq file and key into respective directories

cp /home/jcrain/examples/wheat/fastq/\*.txt.gz $WORKDIR/Illumina/

cp /home/jcrain/examples/wheat/SynOpDH\_UNEAK\_Key\_Multiple\_Fastq.txt $WORKDIR/key/

#This adds the key file and fastq files to the directories created in step 1.

#Step 3 Make tag count file

${TASSEL3} -Xms30G -Xmx30G -fork1 -UFastqToTagCountPlugin -w ${WORKDIR} -e PstI-MspI -s 500000000 -c 1 -endPlugin -runfork1

#This command makes a a tagcount file for each individual by counting the number of tags/sequences that had the barcode for the individual.

#Step 4 Make master tag count file

${TASSEL3} -Xms30G -Xmx30G -fork1 -UMergeTaxaTagCountPlugin -w ${WORKDIR} -c 5 -m 1000000000 -x 10000000 -endPlugin -runfork1

#This command merges the tag count files into one file.

#Step 5

${TASSEL3} -Xms30G -Xmx30G -fork1 -UTagCountToTagPairPlugin -w ${WORKDIR} -e 0.03 -endPlugin -runfork1

########Generate Tags by Taxa TBT file###############

${TASSEL3} -Xms30G -Xmx30G -fork1 -UTagPairToTBTPlugin -w ${WORKDIR} -endPlugin -runfork1

#This command creates the tags/sequences per individual in the key file.

######Generate map info file #####################

${TASSEL3} -Xms30G -Xmx30G -fork1 -UTBTToMapInfoPlugin -w ${WORKDIR} -endPlugin -runfork1

#this command generated the map information data.

##########Generate HapMap file ###############

${TASSEL3} -Xms30G -Xmx30G -fork1 -UMapInfoToHapMapPlugin -w ${WORKDIR}{WORKDIR} -mnMAF 0.05 -mxMAF 0.5 -mnC 0 -mxC 1 -endPlugin -runfork1

#this command generates the hap files, depth count files, and fasta file of the SNPs.

TASSEL V GBSv2 Pipeline:

#Step 1

#have to request memory but aslo specify it calling the plugin

## GBSSeqToTagDBPlugin - RUN Tags to DB require min quality score, 50 base pair tags, and up to 250M kmers in the database

$TASSEL -Xms20G -Xmx20G -fork1 -GBSSeqToTagDBPlugin -e ApeKI \

-i $SEQUENCE \

-db ${NAME}.db \

-k ${KEYFILE} \

-kmerLength 64 -minKmerL 50 -mnQS 20 -mxKmerNum 250000000 \

-endPlugin -runfork1 >> ${NAME}\_pipeline.out

#Builds a database of unique tags found in fastq files.

#Step 2

## TagExportToFastqPlugin - export Tags to align to reference

$TASSEL -fork1 -TagExportToFastqPlugin \

-db ${NAME}.db \

-o ${NAME}\_tagsForAlign.fa.gz -c 10 \

-endPlugin -runfork1 >> ${NAME}\_pipeline.out

#Export tags to align to the reference genome.

#Step 3

## RUN BOWTIE #-S is write to SAM file -U is unparied reads to be aligned -x is aligned files

bowtie2 --end-to-end \

-x ${REFERENCE} \

-U ${NAME}\_tagsForAlign.fa.gz \

-S ${NAME}.sam >> ${NAME}\_pipeline.out

#Align the tags.

#Step 4

## SAMToGBSdbPlugin - SAM to DB, update database with alignment information

$TASSEL -Xms20G -Xmx20G -fork1 -SAMToGBSdbPlugin \

-i ${NAME}.sam \

-db ${NAME}.db \

-aProp 0.0 -aLen 0 \

-endPlugin -runfork1 >> ${NAME}\_pipeline.out

#Update the database with sequence information for the tags.

#Step 5

## DiscoverySNPCaller

$TASSEL -Xms20G -Xmx20G -fork1 -DiscoverySNPCallerPluginV2 \

-db ${NAME}.db \

-mnLCov 0.1 -mnMAF 0.01 -deleteOldData true \

-endPlugin -runfork1 >> ${NAME}\_pipeline.out

#Identify SNPs within the database.

#Step 6

## SNPQualityProfilerPlugin - RUN QUALITY PROFILER

$TASSEL -Xms20G -Xmx20G -fork1 -SNPQualityProfilerPlugin \

-db ${NAME}.db \

-statFile ${NAME}\_SNPqual\_stats.txt \

-endPlugin -runfork1 >> ${NAME}\_pipeline.out

#Get SNP quality information.

#Step 7

## UpdateSNPPositionQualityPlugin - UPDATE DATABASE WITH QUALITY SCORE fast < 30 minutes 15GB

$TASSEL -Xms20G -Xmx20G -fork1 -UpdateSNPPositionQualityPlugin \

-db ${NAME}.db \

-qsFile ${NAME}\_SNPqual\_stats.txt \

-endPlugin -runfork1 >> ${NAME}\_pipeline.out

#Update the database with SNP quality information.

#Ends SNP discovery with database

#Use Production SNP caller to get SNPs and filter

#Step 8

## ProductionSNPCallerPluginV2 - RUN PRODUCTION PIPELINE - output .vcf

$TASSEL -Xms20G -Xmx20G -fork1 -ProductionSNPCallerPluginV2 \

-db ${NAME}.db \

-i ${SEQUENCE} \

-k ${KEYFILE} \

-o ${NAME}.vcf \

-e ApeKI -kmerLength 64 \

-endPlugin -runfork1 >> ${NAME}\_pipeline.out

#Run the production SNP caller.

#Step 9 ## Convert to Hapmap format

$TASSEL -Xms20G -Xmx20G -fork1 -vcf ${NAME}.vcf \

-export ${NAME} -exportType Hapmap >> ${NAME}\_pipeline.out

#Converts VCF file into hapmap file.

##Step 10 #clean up

#make sure to remove unneeded files other files and compress the vcf file