# Activity 2: Running the TASSEL pipeline

In this exercise, you will align GBS sequence tags to a reference genome and call SNPs using the TASSEL “discovery” pipeline. This data is from a strawberry breeding population, and the reference genome is from the strawberry species *Fragaria vesca*.

**TASSEL requires three inputs:**

**1) A barcode key file.** This identifies individual samples by their barcodes and gives their locations on a 96-well plate.

**2) A reference genome fasta file.** This is the sequence to which the sequence tags will be aligned.

**3) One or more Illumina Fastq files representing samples from a GBS population.** These files represent pooled, barcoded samples that were sequenced in one or more lanes on an Illumina sequencer.

**For this exercise, the tab-separated key file is located in the following subdirectory, and is called key.tsv:**

/nfs1/Teaching/data/viningk/GBS/barcode\_key

**The strawberry reference genome fasta file is located in the following subdirectory, and is called Fvb\_genome.fa:**

/nfs1/Teaching/data/viningk/GBS/PhytozomeV10/

**The Illumina fastq file is located in the following subdirectory, and is called SUB\_C1JDAACXX\_s\_2\_fastq.txt:**

/nfs1/Teaching/data/viningk/GBS/Illumina\_fastq/

**Before running the pipeline, look at the key file using the less command. Remember that the –S flag keeps lines from wrapping, and left and right arrow keys can be used to scroll back and forth.**

less -S /nfs1/Teaching/data/viningk/GBS/barcode\_key/key.tsv

Notice that some of the column entries are long enough that they run into adjacent columns and make the file difficult to read. Try the 'column' command to make the file easier to read. The -t flag tells the column utility to determine the number of columns in the file and create a table.

column -t key.tsv

What kind of information is contained in the key file?

**STEP-BY-STEP INSTRUCTIONS FOR RUNNING THE TASSEL PIPELINE**

**Step 1: In your home directory, make a main TASSEL directory.**

mkdir tassel

**Step 2: Change directory to move into the TASSEL directory you just created.**

cd tassel

**Step 3: Within the main TASSEL directory, make subdirectories to hold output files**.

mkdir Illumina

mkdir key

mkdir referencegenome

mkdir tagCounts

mkdir tagsByTaxa

mkdir mergedTagCounts

mkdir topm

mkdir tbt

mkdir mergedTBT

mkdir hapMap

**Step 4: Change directory to the “key” subdirectory, then make a symbolic link to the key file using the "link symbolic" command, ln -s. Use the line list command, ls -l, to see what the symbolic link looks like.**

cd key

ln -s /nfs1/Teaching/data/viningk/GBS/barcode\_key/key.tsv

ls -l

**Step 5: Move up one level from the key subdirectory back to the TASSEL main directory, then move down into the reference genome subdirectory. Make a symbolic link to the strawberry reference genome.**

cd ../referencegenome

ln -s /nfs1/Teaching/data/viningk/GBS/PhytozomeV10/Fvb

**Step 6: Move up one level from the reference genome subdirectory back to the TASSEL main directory, then move down into the Illumina subdirectory. Make a symbolic link to the fastq file with the ln –s command. Then, move back up one level so that you are again in the TASSEL main directory. Hint: Print your current working directory with the “pwd” command to make sure you're in the right place!**

cd ../Illumina

ln -s /nfs1/Teaching/data/viningk/GBS/Illumina\_fastq/SUB\_C1JDAACXX\_s\_2\_fastq.txt

cd ..

**Now you're ready to run TASSEL! Run each plugin from the main TASSEL directory using the commands shown below. \*NOTE: COPY AND PASTE CAREFULLY! It is very important to make sure that each command consists of a single line of text. Mistakes in copying/pasting may result in empty output files that need to be deleted before proceeding. If you make a mistake, check the output subdirectory for empty files.**

**Step 7a. FastqToTagCountPlugin. This counts occurrences of each unique sequence tag in the fastq file. It will output a file in "count" format called SUB\_C1JDAACXX\_s\_2.cnt**

/local/cluster/hts/gbs/tassel3.0\_standalone/run\_pipeline.pl -Xmx512g -fork1 -FastqToTagCountPlugin -i Illumina -k key/key.tsv -e ApeKI -o tagCounts -endPlugin -runfork1

How many barcodes were found in the key file?

How many reads were in the fastq file, and how many of them contained a barcode and cut site overhang?

**Step 7b: MergeMultipeTagCounts plugin. This merges tag counts from multiple .cnt files into a master list. The -c flag specifies that only tags with counts >=5 should be kept. The -t flag tells the plugin to output a fastq format file.**

/local/cluster/hts/gbs/tassel3.0\_standalone/run\_pipeline.pl -Xmx512g -fork1 -MergeMultipleTagCountPlugin -i tagCounts -o mergedTagCounts/FRA\_MasterTags.cnt -c 5 -t -endPlugin -runfork1

**Step 7c: Alignment to the reference genome with bowtie2. The -x flag specifies the reference genome base name. The -U flag specifies the fastq file to be aligned. The -S flag tells bowtie2 to output a sequence alignment map (.sam) file: FRA\_AlignedMasterTags.sam in mergedTagCounts subdirectory.**

bowtie2 -x /nfs1/Teaching/data/viningk/GBS/PhytozomeV10/Fvb -U mergedTagCounts/FRA\_MasterTags.cnt.fq -S mergedTagCounts/FRA\_AlignedMasterTags.sam

**Step 7d: SamConverter plugin. This converts a .sam file into a binary tagsOnPhysicalMap file. Output: 2 files in the topm folder**

**FRA\_MasterTags.topm.bin, FRA\_MasterTags.topm.bin.log**

/local/cluster/hts/gbs/tassel3.0\_standalone/run\_pipeline.pl -fork1 -SAMConverterPlugin -i mergedTagCounts/FRA\_AlignedMasterTags.sam -o topm/FRA\_MasterTags.topm.bin -endPlugin -runfork1

**Step 7e. Change directory into the topm subdirectory and look at the .log file using “less”. How many tags were aligned to unique positions in the reference genome? How many aligned to multiple positions? What happens if you try to look at the .topm.bin file using the “less” command?  
Make sure you cd back up to the main TASSEL directory before proceeding to the next step.**

cd topm

less FRA\_MasterTags.topm.bin.log

cd ..

**Step 7f: Run the MergeMultipleTagCounts plugin again, this time without the -t flag. This will outputs a "count" format .cnt file that is used by the next plugin.**

/local/cluster/hts/gbs/tassel3.0\_standalone/run\_pipeline.pl -Xmx512g -fork1 -MergeMultipleTagCountPlugin -i tagCounts -o mergedTagCounts/FRA\_MasterTags.cnt -c 5 -endPlugin -runfork1

**Step 7g: FastqToTBT plugin. This produces tag counts for each individual in the population. The -t flag indicates a binary “count” format file is being used as input. The -y flag tells the plugin to output specific tag counts (maximum 127) instead of simple presence-absence, 0/1 format.**

/local/cluster/hts/gbs/tassel3.0\_standalone/run\_pipeline.pl -fork1 -FastqToTBTPlugin -i Illumina -k key/key.tsv -e ApeKI -o tbt -t mergedTagCounts/FRA\_MasterTags.cnt -y -endPlugin -runfork1

**Step 7h: MergeTagsByTaxaFiles plugin. This merges separate files produced in the previous step. Since we are only working with one input file, this process will simply rewrite the input file into the output subdirectory.**

/local/cluster/hts/gbs/tassel3.0\_standalone/run\_pipeline.pl -Xmx512g -fork1 -MergeTagsByTaxaFilesPlugin -i tbt -o mergedTBT/SUB\_C1JDAACXX\_s\_2.tbt.byte -endPlugin -runfork1

**Step 7i: TagsToSNPByAlignment plugin. This is the Discovery SNP caller. Run it to see what it does, but our small dataset does not have enough coverage to produce any SNPs. Output from a full data set would be one .txt file per chromosome: e.g. mergedTBT.c1.hmp.txt.**

/local/cluster/hts/gbs/tassel3.0\_standalone/run\_pipeline.pl -Xmx512g -fork1 -TagsToSNPByAlignmentPlugin -i mergedTBT/SUB\_C1JDAACXX\_s\_2.tbt.byte -m topm/FRA\_MasterTags.topm.bin -o hapMap/chr+\_hmp.txt -mnLCov 0.001 -mnMAC 0 -mnMAF 0.0001 -sC 0 -eC 7 -endPlugin -runfork1