**File S1**

**GBS Data Processing Flow**

1. ***Pre-processing sequence data***

The pipeline (process\_radtags) is used to split raw sequence data (a FASTQ file) containing multiple samples based on barcodes, to remove all barcodes and sequencing adapters, to trim reads into 80bp, to filter reads based on Illumina’s Chastity filter.

% $/home/local/bin/**process\_radtags** [-f in\_file | -p in\_dir [-P] | -1 pair\_1 -2 pair\_2] -b barcode\_file -o out\_dir -e enz [-c] [-q] [-r] [-t len] [-D] [-w size] [-s lim] [-h]

The barcode\_file is as follows:

ACCGT

TGCA

……

#See more details of all program options from http://creskolab.uoregon.edu/stacks/.

1. ***Stacks operation***

**2-1** Build tags *de novo*, detect haplotypes in each individual.

% $/home/local/bin/**ustacks** -t file\_type -f file\_path [-d] [-r] [-o path] [-i id] [-m min\_cov] [-M max\_dist] [-p num\_threads] [-R] [-H] [-h]

The output files include 1) XXX.tags.tsv, 2) XXX.snps.tsv, and 3) XXX.alleles.tsv.

**# -m=3, -M=1, -H for haploid genome.**

**2-2** Merge loci from multiple individuals to form a catalog.

% $/home/local/bin/**cstacks** -b batch\_id -s sample\_file [-s sample\_file\_2 ...] [-o path] [-n num] [-g] [-p num\_threads] [--catalog path] [-h]

The catalog are named as batch\_X.catalog.tags.tsv

**# -n=2 for three pine species and -n=1 for one pine species.**

**2-3** Match loci from each individual against the catalog.

% $/home/ local/bin/**sstacks** -b batch\_id -c catalog\_file -s sample\_file [-r sample\_file] [-o path] [-p num\_threads][-g] [-x] [-v] [-h]

**2-4** Call SNPs from multiple individuals.

% $/home/ local/bin /**populations** -b batch\_id -P path -M path [-r min] [-m min] [-B blacklist] [-W whitelist] [-s] [-e renz] [-t threads] [-v] [-h]

**#** **-m=3 and -r=0.5 in our study.**

#See more details of all program options from <http://creskolab.uoregon.edu/stacks/>

1. ***Bowtie2-GATK operation***

**3-1** Construct reference sequences based on the catalog of consensus loci (batch\_X.catalog.tags.tsv).

The Perl script (**reference\_generated\_M1.pl**) is only applicable for haploid genomes, and processed as follows: 1) filter loci with > 50% missing genotype and 2) filter the loci as repeats in which > 2 individuals have ≥ 2 genotypes.

% $home/GATK-analyses/perl reference\_generated\_M1.pl

A list file containing the SNP file names of all analyzed individuals in this step. For example,

Indv\_1.snps.tsv

Indv\_2.snps.tsv

……

A variance ($ind\_number\_th) needs to be defined. The value refers to the half number of all analyzed individuals.

**3-2** Align reads of each individual to the reference.

**3-2-1** Index reference sequences.

% $java -jar /home/picard-tools-1.97/**CreateSequenceDictionary.jar** R=reference.fasta O=reference.dict

% $**samtools** **faidx** reference.fasta

% $**bowtie2 -build** reference.fasta reference

**3-2-2** Alignment

% $/home/bowtie2-2.1.0/**bowtie2** -x reference\_file -U fastqfile(XXX.fastq) -D 20 -R 3 -N 0 -L 20 -i S,1,0.50 -S outputfile (XXX.sam)

**3-2-3** SAM files transform into BAM files

% $**samtools view** -bS $output/$name.sam >>$output/$name.bam

% $**samtools sort** $output/$name.bam $output/$name.sorted

% $java -jar /home/picard-tools-1.97/**ReorderSam.jar** I=$output/$name.sorted.bam O=$output/$name.A.bam R=$reference.fasta

% $java -jar /home/ picard-tools-1.97/**AddOrReplaceReadGroups.jar** I=$output/$name.A.bam O=$output/$name.B.bam RGID=$name RGLB=L5 RGPL=illumina RGPU=80bp RGSM=$name RGDS=contig

% $**samtools index** $output/$name.B.bam

**3-3** Call SNPs in GATK

% $java -jar /home/GenomeAnalysisTK-2.7-2-g6bda569/**GenomeAnalysisTK.jar** -T UnifiedGenotyper -R $reference -I ./GATK/ indv\_1.B.bam -I ./GATK/ indv\_2.B.bam … … -ploidy 1 -glm BOTH -o $output

1. ***Extract common and specific SNPs between two vcf files from Stacks and GATK***

**4-1** Transform position information in the stacks.vcf

% $perl **transform-for-stacks.pl**

**4-2** Extract common SNPs

% $/home/Bedtools/**IntersectBed** –a stacks.vcf –b GATK.vcf –c >stacks\_common.vcf

% $/home/Bedtools/**IntersectBed** –b stacks.vcf –a GATK.vcf –c >GATK\_common.vcf

**4-3** Extract specific SNPs

% $/home/Bedtools/**IntersectBed** –a stacks.vcf –b GATK.vcf –wa > stacks\_specific.vcf

% $/home/Bedtools/**IntersectBed** –a stacks.vcf –b GATK.vcf –wb > GATK\_specific.vcf

1. **Filter and combine two sets of SNPs**

**5-1** combining common SNPs based on the D-value. If SNPs with >2 D-value, genotypes in *Stacks* are retained; if not, genotypes in both are merged.

% $perl **unified\_common\_SNPs.pl**

**5-2** Remove non-reference SNPs from specific SNPs of stacks based on the reference file.

% $perl **filter-specific-stacks.pl**

**5-3** Filter SNPs with >2 D-value or AC=1 from specific SNPs of GATK.

% $perl **filter-specific-GATK.pl**

1. **Filter SNPs of the overlapping regions between reference sequences**

**6-1** Identify overlapping regions between two reference sequences.

% $**velveth** output\_directory/ 21 –fasta –short reference.fasta

% $**velvetg** output\_directory/

The output file is contigs.fa

% $perl **filter\_contigs.pl**

% $perl **filter\_snp\_contigs\_2.pl**

The Perl scripts are used to identify the correct contigs, which of the beginning and terminal sequences must be same as the residual sites of restriction enzyme, and of the overlapping regions between two reference sequences have ≤ 2bp mismatches.

**6-2** Filter one of SNPs called twice in the overlapping regions

% $perl **snp\_filter.pl**

% $perl **snp\_filter\_1.pl**

This two Perl scripts are used to filter one of SNPs within the overlapping regions from the combined SNPs file based on the position information within reference sequences.