**Coding Supplement**

# trim adaptors

trim\_galore -q 20 --paired --illumina file1.fastq file2.fastq

# align reads to features

star --runThreadN 4 --genomeDir /genome\_dir --readFilesIn file1.fastq file2.fastq --outFileNamePrefix file"

# count reads across features

featureCounts -p -a file.gtf -B -C -G file.fasta -s 2 -T 4 -o files.bam

# haplotype caller

java -Xmx10g -jar GenomeAnalysisTK.jar -T HaplotypeCaller -drf DuplicateRead -R file.fasta -I file.bam -dontUseSoftClippedBases -stand\_call\_conf 20.0 -nct 8 -o file.g.vcf

# merge vcfs

vcf-merge file1.g.vcf.gz file2.g.vcf.gz > merged\_raw\_variants.vcf

# select snps

java -jar GenomeAnalysisTK.jar -T SelectVariants -R file.fasta -V raw\_variants.vcf -selectType SNP -o raw\_snps.vcf

# filter snps

java -jar GenomeAnalysisTK.jar -T VariantFiltration -R file.fasta -V raw\_snps.vcf --filterExpression "QD < 2.0 || FS > 60.0 || MQ < 40.0 || MQRankSum < -12.5 || ReadPosRankSum < -8.0" --filterName "standard\_snp\_filter" -o filtered\_snps.vcf

# filter minor allele frequency 5% and max missing genotype 90%

vcftools --vcf filtered\_snps.vcf --maf 0.05 --max-missing 0.9 --recode --out maf\_0.5\_maxmiss\_0.9\_snps"

# phase

java -jar GenomeAnalysisTK.jar -T ReadBackedPhasing -R file.fasta -I file.bam --variant snps.vcf -o phased.vcf --phaseQualityThresh 20.0

# count allele reads

java -jar GenomeAnalysisTK.jar -R file.fasta -T ASEReadCounter -U ALLOW\_SEQ\_DICT\_INCOMPATIBILITY -o allele\_counts.csv -I file.bam -sites phased.vcf

# output snp genotypes

java -jar GenomeAnalysisTK.jar -R file.fasta -T VariantsToTable -V phased.vcf -F CHROM -F POS -GF GT -GF HP -o snp\_table.txt

# Basic DESeq2 design

dds <- DESeqDataSetFromMatrix(countData = cts,

colData = colData,

design= ~species)

dds <- estimateSizeFactors(dds)

idx <- rowSums(counts(dds, normalized=TRUE) >= 10 ) >= (nrow(colData))

dds <- dds[idx,]

dds <- DESeq(dds)