Table S3: Related bioinformatic scripts in our research

1. **Transcript reconstruction and quantification:**

stringtie -p 10 -G genome.gtf -o [output\_gtf\_file] [input\_bam\_file]

stringtie --merge -p 10 -G genome.gff [output\_gtf\_file1] [output\_gtf\_file2] .. [output\_gtf\_file9] -o total.gtf

stringtie -e -p 10 -G total.gtf -o output\_merge.gtf [input\_bam\_file]

1. **Identification of differential expressed genes**

> library(DESeq2)

> database <- as.matrix(read.csv("transcript\_count\_matrix.csv", row.names="transcript\_id"))

> condition <- factor(c("control","control","KD","KD"))

> coldata <- data.frame(row.names = colnames(database), condition)

> countData <- countData[, rownames(colData)]

> dds <- DESeqDataSetFromMatrix(countData = countData, colData = colData, design = ~ condition)

> dds <- DESeq(dds)

> res <- results(dds)

> resordered <- res[order(res$padj),]

> summary(res)

> write.csv(as.data.frame(resordered), file="results.csv")

1. **SNP calling**

# UnifiedGenotyper

# output read-$i.snv-gatk-snp.vcf

&execsys("java -Xmx2G -jar $cfg{gatk\_dir}/GenomeAnalysisTK.jar -T UnifiedGenotyper -R $cfg{genome} -I read-$i.recal.bam -D known\_snp.conv -A AlleleBalance -stand\_call\_conf 50.0 -stand\_emit\_conf 10.0 -dcov 200 -glm SNP -out\_mode EMIT\_VARIANTS\_ONLY -log log.UnifiedGenotyper-snp.$i -o read-$i.snv-gatk-snp.vcf");

# VariantFiltration

# filtering read-$i.snv-gatk-snp.vcf to get read-$i.snv-gatk-snp.fil.vcf

&execsys("java -Xmx2G -jar $cfg{gatk\_dir}/GenomeAnalysisTK.jar -T VariantFiltration -R $cfg{genome} -V read-$i.snv-gatk-snp.vcf --clusterWindowSize 10 --filterExpression \"QUAL < 30.0 || QD < 5.0\" --filterName \"HARD\_TO\_VALIDATE\" -log log.VariantFiltration-snp.$i -o read-$i.snv-gatk-snp.fil.vcf");

&execsys("java -Xmx2G -jar $cfg{gatk\_dir}/GenomeAnalysisTK.jar -T UnifiedGenotyper -R $cfg{genome} -I read-$i.recal.bam -D known\_snp.conv -A AlleleBalance -stand\_call\_conf 50.0 -stand\_emit\_conf 10.0 -dcov 200 -glm INDEL -out\_mode EMIT\_VARIANTS\_ONLY -log log.UnifiedGenotyper-indel.$i -o read-$i.snv-gatk-indel.vcf");

&execsys("java -Xmx2G -jar $cfg{gatk\_dir}/GenomeAnalysisTK.jar -T VariantFiltration -R $cfg{genome} -V read-$i.snv-gatk-indel.vcf --clusterWindowSize 10 --filterExpression \"QUAL < 10.0\" --filterName \"HARD\_TO\_VALIDATE\_1\" --filterExpression \"MQ0 >= 4 && ((MQ0 / (1.0 \* DP)) > 0.1)\" --filterName \"HARD\_TO\_VALIDATE\_2\" -log log.VariantFiltration-indel.$i -o read-$i.snv-gatk-indel.fil.vcf");

# CombineVariants

&execsys("java -Xmx2G -jar $cfg{gatk\_dir}/GenomeAnalysisTK.jar -T CombineVariants -R $cfg{genome} --variant read-$i.snv-gatk-snp.fil.vcf --variant read-$i.snv-gatk-indel.fil.vcf -o read-$i.snv-gatk.vcf.tmp -log log.CombineVariants.$i");

1. **Calculation of N50**

import sys

a = sys.argv[1]

seq = ''

length = []

for line in open(a):

if line.startswith('>') and seq == '':

id = line.split()[0]

elif line[0] != '>':

seq = seq +line

elif line.startswith('>') and seq != '':

leng = len(seq)

length.append(leng)

seq = ''

leng = len(seq)

length.append(leng)

length.sort()

length.reverse()

total = sum(length)

a = 0

for x in range(len(length)):

a += length[x]

if float(a)/float(total) >= 0.5:

print (length[x])

break

1. **The selection of the longest transcript**

**#coding=gbk**

import sys

input = sys.argv[1]

seq = ''

ac\_list = []

dict = {}

length = []

for line in open(input):

if line.startswith('>') and seq == '':

id = line.split()[0]

if line[0] != '>':

seq = seq + line.strip()

if line.startswith('>') and seq != '':

dict[id]=seq

seq =''

id = line.split()[0]

dict[id]=seq

uniq = {}

for k,v in dict.items():

name = k

v = [v]

if name not in uniq:

uniq[name] = v

else:

uniq[name] += v

max\_seq = {}

for k,v in uniq.items():

seq = max(v,key = len)

max\_seq[k] = seq

for k,v in max\_seq.items():

if len(v) >= 300:

print k+'\n'+v