ls

######Gene\_shift run

perl ../../../HiSeqTool/runHTseqV3\_shift\_ccrinas Mus\_musculus UCSC mm9 120816\_SN371\_0110\_BD15CFACXX Sun\_Luzhe\_07272012 gene\_id > Sun120816HTseq\_geneShift.log 2>&1 & perl ../../../HiSeqTool/runHTseqV3\_shift\_matlab Mus\_musculus UCSC mm9 130219\_SN371\_0122\_AC167DACXX Sun\_Luzhe\_02072013 gene\_id > Sun130219HTseq\_geneShift.log 2>&1 &

######

######Builg gene\_sift file



###

###1\_step. File download 1)Gene GTF file

1. All introns in bed format from UCSC table browser
2. All intergenic regions in bed format from UCSC table browser 4)allMrna from UCSC databasae

5)ensGene from UCSC databasae 6)pseudoYale60Gene from UCSC databasae 7)vegaGene from UCSC databasae 8)xenoMrna from UCSC databasae 9)xenoRefGene from UCSC databasae

###2\_step. Process gene-exon, intron and intergenic region and make the clear no-gene region 1)Merge intron and intergenic region as 'gene-free-region-1'

1. remove regions, which is overlaping with exons and region from files '4)~10)', from gene-free-region-1

#'4)~10)' are optional, so far we just check the exon regions from gene.gtf file.

1. Trim 100 bps from intron regions and 1000 bps from integenic regions by both ends as 'GFRM'

###3\_step. Shift gene exons to GFRM

1. shift exons to the closest gene free region by direction "R-L-R-L..."
2. sort exons by chromosome

###

##Program running example:

###

####################################################################

perl ../../GEFRshift.pl

-G ../mm10\_genes.gtf

-I ../mm10\_UCSC\_Introns.tsv

-T ../mm10\_UCSC\_Intergenic.tsv

-m ../otherRefGene/mm10\_UCSC\_allMrna.bed

-x ../otherRefGene/mm10\_UCSC\_xenoMrna.bed

-z ../otherRefGene/mm10\_UCSC\_xenoRefGene.bed

-e ../otherRefGene/mm10\_UCSC\_ensGene.bed

-p ../otherRefGene/mm10\_UCSC\_pseudoYale60Gene.bed

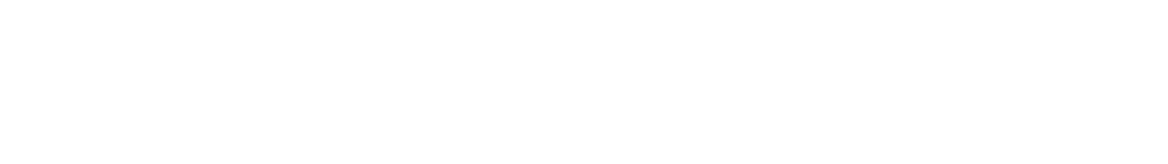
-v ../otherRefGene/mm10\_UCSC\_vegaGene.bed

> mm10\_exonShift.log 2>&1 &

####################################################################

###Process steps=============================================================

###1\_step. Clean the gene, intron and integenic file from Chrom1\_22 and X, Y



###

#check unique chromosome in the file; Not necessary cut -f 1 hg19\_UCSC\_Intron.tsv | sort | uniq -c

cut -f 1 genes.gtf | sort | uniq -c

cut -f 1 hg19\_UCSC\_Intergenic.tsv | uniq -c

#Remove genes with \_random, \_hap1, \_hap2; Not necessary

grep -P -v '\_random\t' genes.gtf | grep -P -v "^chr.\*hap.\t" | grep -P -v "^chrUn\_gl.\*\t" > hg19\_genes.gtf

grep -P -v '\_random\t' hg19\_UCSC\_Introns.tsv | grep -P -v "^chr.\*hap.\t" | grep -P -v "^chrUn\_gl.\*\t" > hg19\_UCSC\_Introns.bed

grep -P -v '\_random\t' hg19\_UCSC\_Intergenic.tsv | grep -P -v "^chr.\*hap.\t" | grep -P -v "^chrUn\_gl.\*\t" > hg19\_UCSC\_Integenic.bed

#Re-check cleaned refFlat file; Not necessary cut -f3 hg19\_genes.gtf | sort | uniq -c

cut -f1 hg19\_UCSC\_Intron.bed | sort | uniq -c cut -f1 hg19\_UCSC\_Integenic.bed | sort | uniq -c

###

##2\_step. Process gene-exon, intron and intergenic region and make the clear no-gene region

###

#Get exon only file, sort and combine overlap region

grep -P '\texon\t' hg19\_genes.gtf | sort -k1,1 -k4,4n -k5,5n > hg19\_geneExon.gtf

grep -P '\texon\t' hg19\_genes.gtf | cut -f1,4,5 | sort -k1,1 -k2,2n -k3,3n > hg19\_exon.bed bedtools merge -i hg19\_exon.bed -n > hg19\_exon\_merge.bed

#Sort and merge overlap intron region

sort -k1,1 -k2,2n -k3,3n hg19\_UCSC\_Introns.bed > hg19\_UCSC\_Introns\_sort.bed bedtools merge -i hg19\_UCSC\_Introns\_sort.bed -nms > hg19\_UCSC\_Introns\_merge.bed sed 's/uc.\*/intron/' hg19\_UCSC\_Introns\_merge.bed > hg19\_UCSC\_Introns\_merged.bed

#combine meged intron and integenic file

cat hg19\_UCSC\_Introns\_merged.bed hg19\_UCSC\_Integenic.bed > hg19\_UCSC\_introInter.bed sort -k1,1 -k2,2n -k3,3n hg19\_UCSC\_introInter.bed > hg19\_UCSC\_introInter\_sort.bed

#remove the region overlapping exon regions from hg19\_UCSC\_introInter\_sort.bed bedtools subtract -a hg19\_UCSC\_introInter\_sort.bed -b hg19\_exon\_merge.bed > hg19\_UCSC\_GeneFreeRegion.bed

###

#optional: bedtools complement [OPTIONS] -i <bed/gff/vcf> -g <genome>

###

###

###3\_step. R process to calculate gene-free region length, cut region ends by intron 100 and

integenic region 1000 bps, Done by shell command

###

#Calculate gene-free region length read.delim(file="hg19\_UCSC\_GeneFreeRegion.bed", sep="\t", header=F) -> hg19\_GFR

#Minus GeneFreeRegion 100bp from both end for intron region

#Minus GeneFreeRegion 1000bp from both end for intergenic region hg19\_GFR -> hg19\_GFRM

hg19\_GFRM[hg19\_GFRM$V4=='intron',2] <- hg19\_GFR[hg19\_GFRM$V4=='intron',2] + 100 hg19\_GFRM[hg19\_GFRM$V4!='intron',2] <- hg19\_GFR[hg19\_GFRM$V4!='intron',2] + 1000 hg19\_GFRM[hg19\_GFRM$V4=='intron',3] <- hg19\_GFR[hg19\_GFRM$V4=='intron',3] - 100 hg19\_GFRM[hg19\_GFRM$V4!='intron',3] <- hg19\_GFR[hg19\_GFRM$V4!='intron',3] - 1000

#compute the fragment size

hg19\_GFRM[,5] <- hg19\_GFRM[,3] - hg19\_GFRM[,2]

write.table(hg19\_GFRM, file="hg19\_GFRM.bed", sep="\t", col.names=F, row.names=F, quote=F)

####

####4. Comapre hg19\_geneExon.gtf with hg19\_GFRM.bed and shif the exon to the closest gene free region

####

mkdir hg19ExonShift

sort -k1,1 -k2,2n -k3,3n hg19\_GFRM.bed > hg19\_GFRM\_sort.bed input A: exon only gtf file <- hg19\_geneExon.gtf

input B: gene free inton or integenic region <- hg19\_GFRM\_sort.bed

#Get exon shift data

perl exonFreeRegionShift.pl -EX hg19\_geneExon.gtf -FR hg19\_GFRM\_sort.bed output: hg19\_geneExon\_shift.gtf

####

####Reference 1. Gene table

####

genes.gtf from server

####

####Reference 2. how to get a BED file of all introns

####

1. Go to the UCSC table browser.
2. Select desired species and assembly
3. Select group: Genes and Gene Prediction Tracks
4. Select track: UCSC Genes (or Refseq, Ensembl, etc.)
5. Select table: knownGene
6. Select region: genome (or you can test on a single chromosome or smaller region)
7. Select output format: BED - browser extensible data
8. Enter output file: UCSC\_Introns.tsv
9. Select file type returned: gzip compressed
10. Hit the 'get output' button
11. A second page of options relating to the BED file will appear.
12. Under 'create one BED record per:'. Select 'Introns plus'
13. Add desired flank for introns being returned, or leave as 0 to get just the introns
14. Hit the 'get BED' option

####

####Reference 3. how to get a BED file of integenic regions

####

1. [Go UCSC Table Browser (http://genome.ucsc.edu/cgi-bin/hgTables)](http://genome.ucsc.edu/cgi-bin/hgTables))
2. select a gene set from the "Genes and Gene prediction tracks"
3. choose "output format: selected fields from primary and related tables."
4. enter a name in the "output file" box. Hit "get output,"
5. on the next page, choose the chrom, txStart and txEnd fields. This should get you a BED file of gene regions that don't have any intron/exon structure.
6. Now go back to the Table Browser and hit the "add custom tracks" button. Upload the file of gene regions you just created.
7. Select the new custom track of gene regions in the Table Browser.
8. Hit the "intersection: create" button, and select the custom track again (that is, create an intersection of the custom track with itself).
9. Select the option for "Base-pair-wise intersection (AND) of User Track and User Track."
10. At the bottom of the page, select both checkboxes next to the "Complement User Track before base-pair-wise intersection/union" options.
11. You can select "output format: BED - browser extensible data" to get the output as a list of coordinates.

This output should be all of the intergenic regions.

####

####Reference43. other tables:

####xenorRefGene: those genes are not shown in gene.gtf, but thery are aligned though other species and found in our data

###mm9

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from mm9.xenoRefGene" > mm9\_UCSC\_xenoRefGene.tsv

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select tName, tStart, tEnd from mm9.xenoMrna" > mm9\_UCSC\_xenoMrna.tsv

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select tName, tStart, tEnd from mm9.all\_mrna" > mm9\_UCSC\_allMrna.tsv

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from mm9.pseudoYale60" > mm9\_UCSC\_pseudoYale60Gene.tsv

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from mm9.vegaGene" > mm9\_UCSC\_vegaGene.tsv

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from mm9.ensGene" > mm9\_UCSC\_ensGene.tsv

#hg18

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from hg18.xenoRefGene" > hg18\_UCSC\_xenoRefGene.tsv

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select tName, tStart, tEnd from hg18.all\_mrna" > hg18\_UCSC\_allMrna.tsv

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select tName, tStart, tEnd from hg18.xenoMrna" > hg18\_UCSC\_xenoMrna.tsv

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from hg18.pseudoYale60" > hg18\_UCSC\_pseudoYale60Gene.tsv

#no pseudo gene in hg18

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from hg18.vegaGene" > hg18\_UCSC\_vegaGene.tsv

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from hg18.ensGene" > hg18\_UCSC\_ensGene.tsv

#hg19

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select tName, tStart, tEnd from hg19.all\_mrna" > hg19\_UCSC\_allMrna.bed

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select tName, tStart, tEnd from hg19.xenoMrna" > hg19\_UCSC\_xenoMrna.bed

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from hg19.xenoRefGene" > hg19\_UCSC\_xenoRefGene.bed

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from hg19.pseudoYale60" > hg19\_UCSC\_pseudoYale60Gene.bed

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from hg19.vegaGene" > hg19\_UCSC\_vegaGene.bed

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from hg19.ensGene" > hg19\_UCSC\_ensGene.bed

###mm10

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from mm10.xenoRefGene" > mm10\_UCSC\_xenoRefGene.tsv

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select tName, tStart, tEnd from mm10.xenoMrna" > mm10\_UCSC\_xenoMrna.tsv

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select tName, tStart, tEnd from mm10.all\_mrna" > mm10\_UCSC\_allMrna.tsv

#no mm10\_UCSC\_pseudoYale60Gene.tsv

#mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from mm10.pseudoYale60" > mm10\_UCSC\_pseudoYale60Gene.tsv

#no mm10\_UCSC\_vegaGene.tsv

#mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from mm10.vegaGene" > mm10\_UCSC\_vegaGene.tsv

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from mm10.ensGene" > mm10\_UCSC\_ensGene.tsv

#### hg38

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select tName, tStart, tEnd from hg38.all\_mrna" > hg38\_UCSC\_allMrna.bed

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select tName, tStart, tEnd from hg38.xenoMrna" > hg38\_UCSC\_xenoMrna.bed

mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from hg38.xenoRefGene" > hg38\_UCSC\_xenoRefGene.bed

# no pseudo gene

#mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from hg38.pseudoYale60" > hg38\_UCSC\_pseudoYale60Gene.bed

# no vega gene

#mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from hg38.vegaGene" > hg38\_UCSC\_vegaGene.bed

# no ens gene

#mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e \ "select chrom, txStart, txEnd from hg38.ensGene" > hg38\_UCSC\_ensGene.bed