

```

#!/usr/local/perl -w
@filenames=("GROseq_WT_cortex_rep1","GROseq_WT_cortex_rep2","GROseq_R106W_cortex_rep1","GROseq_R106W_cortex_rep2");
foreach $names (@filenames)
{
open(OUA,">../Count_all_genes/GROseq/$names\_GROseq_body_allgene_sense.txt")||die("Can't write OUB:$!\n");
print "Working on sample: $names\n";my $a1=$a2=$a3=$a4=0;

@chromo=("chr1","chr2","chr3","chr4","chr5","chr6","chr7","chr8","chr9","chr10","chr11","chr12","chr13","chr14","chr15","chr16","chr17","chr18","chr19","chrX","chrY");
foreach $chromosome (@chromo)
{
print "Working on chromosome: $chromosome\n";
## + strand genes
open(INA,"/media/RNA/Genome/Ensembl/exon_intron/Mus_musculus.GRCm38.75.gtf.gene")||die("Can't open INB:$!\n");
open(INB,"../Count_all_genes/GROseq/$names\_plus.bedGraph")||die("Can't open INC:$!\n");
open(INC,"/media/RNA/Genome/Ensembl/exon_intron/Mus_musculus.GRCm38.75.gtf.gene")||die("Can't open IND:$!\n");

## - strand genes
open(IND,"/media/RNA/Genome/Ensembl/exon_intron/Mus_musculus.GRCm38.75.gtf.gene")||die("Can't open INB:$!\n");
open(INE,"../Count_all_genes/GROseq/$names\_minus.bedGraph")||die("Can't open INC:$!\n");
open(INF,"/media/RNA/Genome/Ensembl/exon_intron/Mus_musculus.GRCm38.75.gtf.gene")||die("Can't open IND:$!\n");

### working on the + strand genes
while(<INA>){chomp; if(/^(\w\w?)\t(\w+)\tgene\t(\d+)\t(\d+)\t([+-])\t(ENSMUSG\d+)\t([\w\.-\(\)]+)$/){ if($chromosome eq "chr$1") ## same chromosome
{ $a1++; print "Processes gene lines: $a1\n" if($a1%5000==0); if($5 eq "+")
{$start=$3-1000; $end=$4+1000; for($i=$start;$i<=$end;$i++){ $id="chr$1_$i"; $plus{$id}=0; } } }else{print "error1\t$_\n";}}

while(<INB>){ chomp; if(/^(\w\w?)\t(\d+)\t(\d+)\t(\d+)\t$/){ if($chromosome eq "chr$1"){ $total++; print "Processes lines: $total\n" if($total%1000000==0);
for($i=$2+1;$i<=$3;$i++){ $id="chr$1_$i"; if(exists $plus{$id}){$plus{$id}+=$4; } } }

while(<INC>){ chomp; if(/^(\w\w?)\t(\w+)\tgene\t(\d+)\t(\d+)\t([+-])\t(ENSMUSG\d+)\t([\w\.-\(\)]+)$/){ if($chromosome eq "chr$1"){ $a2++;
if($5 eq "+") { print OUA "$6_$7"; $start=$3-1000; $end=$3-1; for($i=$start;$i<=$end;$i++){ $id="chr$1_$i"; print OUA "\t$plus{$id}"; } for($j=0;$j<1000;$j++)
{ $start_bin=int( (($3 + (($4-$3+1)/1000)*$j)*10 +5)/10 ); $end_bin= int( (($3 + (($4-$3+1)/1000)*($j+1))*10 +5)/10 )-1;
if($end_bin<$start_bin){ $end_bin=$start_bin; } $total_bin=0; for($i=$start_bin;$i<=$end_bin;$i++){ $id_tmp="chr$1_$i"; $total_bin+=$plus{$id_tmp}; }
$mean_bin=int(100*$total_bin/($end_bin-$start_bin+1)+5)/100; print OUA "\t$mean_bin"; }
$start=$4+1; $end=$4+1000; for($i=$start;$i<=$end;$i++){ $id="chr$1_$i"; print OUA "\t$plus{$id}"; } print OUA "\n"; } } }else{print "error3\t$_\n";}}
undef %plus;

while(<IND>){ chomp; if(/^(\w\w?)\t(\w+)\tgene\t(\d+)\t(\d+)\t([+-])\t(ENSMUSG\d+)\t([\w\.-\(\)]+)$/){ if($chromosome eq "chr$1"){
$a3++; print "Processes gene lines: $a1\n" if($a1%5000==0);if($5 eq "-") { $right=$4+1000;
$left=$3-1000; for($i=$left;$i<=$right;$i++){ $id="chr$1_$i"; $minus{$id}=0; } } }
else{print "error4\t$_\n";}}

while(<INE>){ chomp; if(/^(\w\w?)\t(\d+)\t(\d+)\t(\d+)\t$/){ if($chromosome eq "chr$1"){ $total++; print "Processes lines: $total\n" if($total%1000000==0);
for($i=$2+1;$i<=$3;$i++){ $id="chr$1_$i";if(exists $minus{$id}){$minus{$id}+=$4; } } }

while(<INF>){chomp; if(/^(\w\w?)\t(\w+)\tgene\t(\d+)\t(\d+)\t([+-])\t(ENSMUSG\d+)\t([\w\.-\(\)]+)$/){ if($chromosome eq "chr$1"){ $a4++;
if($5 eq "-") { print OUA "$6_$7"; $start=$4+1; $end=$4+1000; for($i=$end;$i>=$start;$i--){ $id="chr$1_$i"; print OUA "\t$minus{$id}"; }
for($j=0;$j<1000;$j++) { $end_bin=int( (($4 - (($4-$3+1)/1000)*$j)*10 +5)/10 ); $start_bin= int( (($4 - (($4-$3+1)/1000)*($j+1))*10 +5)/10 )+1;
if($end_bin<$start_bin){ $end_bin=$start_bin; } $total_bin=0;for($i=$start_bin;$i<=$end_bin;$i++){ $id_tmp="chr$1_$i"; $total_bin+=$minus{$id_tmp}; }
$mean_bin=int(100*$total_bin/($end_bin-$start_bin+1)+5)/100; print OUA "\t$mean_bin"; }
$start=$3-1000; $end=$3-1; for($i=$end;$i>=$start;$i--){ $id="chr$1_$i"; print OUA "\t$minus{$id}"; } print OUA "\n"; } } }else{print "error6\t$_\n";}}
undef %minus;
close INC;
close INA; close INB; close IND;
}
close OUA;
print "Total-gene: $a1 $a2 $a3 $a4\n";
}

```

Supplemental Figure S9. Scripts used for bin based counting of sequencing signals. Perl script used to count bin based sequencing signals in gene bodies for ChIP-seq, ATAC-seq, GRO-seq, mC, and hmC data sets. Counting GRO-seq signals was used as an example.