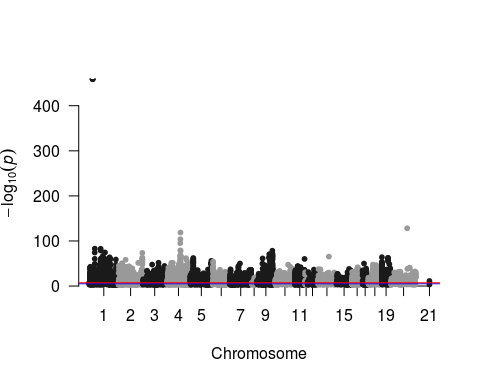
Gene Ranking Analysis

This is the report and documentation of gene analysis for mouse.

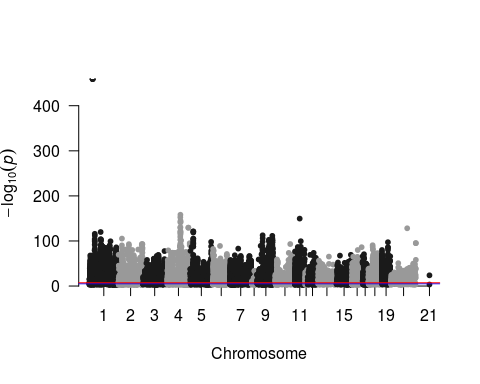
# NGS Data

## All genes without any filtering.

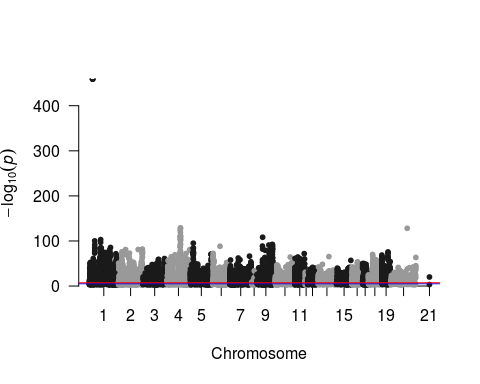
Using Mean -log(p value). Manhattan plot is created for each gene.



Using Max -log(p value). Manhattan plot is created for each gene.

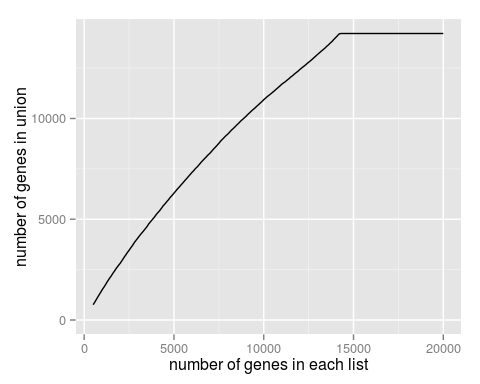


Using Top Quartile Mean -log(p value). Manhattan plot is created for each gene.



The type of gene in above analysis

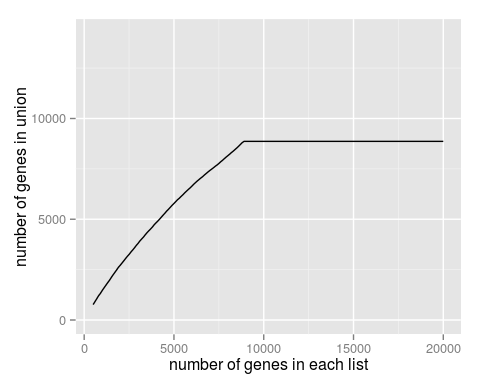
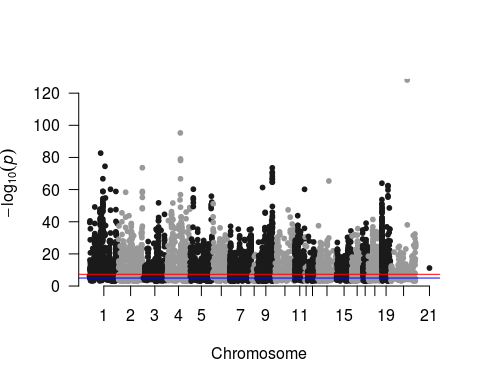
## [1] "3prime\_overlapping\_ncrna"   
## [2] "antisense"   
## [3] "IG\_C\_gene"   
## [4] "IG\_J\_gene"   
## [5] "IG\_LV\_gene"   
## [6] "lincRNA"   
## [7] "miRNA"   
## [8] "misc\_RNA"   
## [9] "polymorphic\_pseudogene"   
## [10] "processed\_pseudogene"   
## [11] "processed\_transcript"   
## [12] "protein\_coding"   
## [13] "pseudogene"   
## [14] "rRNA"   
## [15] "sense\_intronic"   
## [16] "sense\_overlapping"   
## [17] "snoRNA"   
## [18] "snRNA"   
## [19] "TEC"   
## [20] "transcribed\_processed\_pseudogene"   
## [21] "transcribed\_unprocessed\_pseudogene"  
## [22] "TR\_C\_gene"   
## [23] "TR\_D\_gene"   
## [24] "TR\_J\_gene"   
## [25] "TR\_J\_pseudogene"   
## [26] "TR\_V\_gene"   
## [27] "TR\_V\_pseudogene"   
## [28] "unitary\_pseudogene"   
## [29] "unprocessed\_pseudogene"



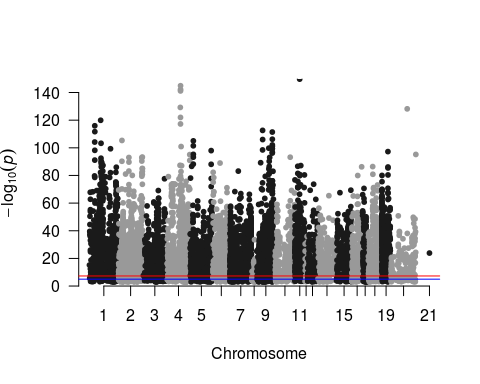
## Filtering for Protein Coding Genes only

Using Mean -log(p value). Manhattan plot is created for each gene.

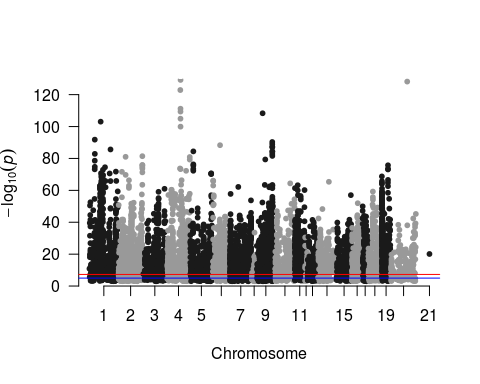
## [1] 8863 11

Using Max -log(p value). Manhattan plot is created for each gene.



Using Top Quartile Mean -log(p value). Manhattan plot is created for each gene.



Union of all top 500 protein coding genes using max, mean, topQ stats. Number of genes in Union Set are

## [1] 1959

## Input contains no \n. Taking this to be a filename to open  
## File opened, filesize is 0.000802 GB.  
## Memory mapping ... ok  
## Detected eol as \n only (no \r afterwards), the UNIX and Mac standard.  
## Looking for supplied sep ',' on line 30 (the last non blank line in the first 'autostart') ... found ok  
## Found 24 columns  
## First row with 24 fields occurs on line 1 (either column names or first row of data)  
## 'header' changed by user from 'auto' to TRUE  
## Count of eol after first data row: 1978  
## Subtracted 1 for last eol and any trailing empty lines, leaving 1977 data rows  
## Type codes ( first 5 rows): 444441133131144333333334  
## Type codes (+ middle 5 rows): 444441133131144333333334  
## Type codes (+ last 5 rows): 444441133131144333333334  
## Type codes: 444441133131144333333334 (after applying colClasses and integer64)  
## Type codes: 444441133131144333333334 (after applying drop or select (if supplied)  
## Allocating 24 column slots (24 - 0 dropped)  
## 0.000s ( 1%) Memory map (rerun may be quicker)  
## 0.000s ( 1%) sep and header detection  
## 0.001s ( 12%) Count rows (wc -l)  
## 0.000s ( 1%) Column type detection (first, middle and last 5 rows)  
## 0.000s ( 0%) Allocation of 1977x24 result (xMB) in RAM  
## 0.008s ( 85%) Reading data  
## 0.000s ( 0%) Allocation for type bumps (if any), including gc time if triggered  
## 0.000s ( 0%) Coercing data already read in type bumps (if any)  
## 0.000s ( 0%) Changing na.strings to NA  
## 0.010s Total

## [1] 1

The Venn Diagram of all the three lists. NGS, 80k and RNA 