main_r

September 8, 2021

1 eQTL boxplot

This is script ported from python to fix unknown plotting error.

[1]:
```
suppressPackageStartupMessages({
    library(tidyverse)
    library(ggpubr)
})
```

1.1 Functions

[2]:
```
feature = "genes"
```

1.1.1 Cached functions

[3]:
```
get_residualized_df <- function(){
    expr_file = "../../_m/genes_residualized_expression.csv"
    return(data.table::fread(expr_file) %>% column_to_rownames("gene_id"))
}
memRES <- memoise::memoise(get_residualized_df)

get_biomart_df <- function(){
    biomart = data.table::fread("../_h/biomart.csv")
}
memMART <- memoise::memoise(get_biomart_df)

get_pheno_df <- function(){
    phenotype_file = paste0('/ceph/projects/v4_phase3_paper/inputs/',
                            'phenotypes/_m/merged_phenotypes.csv')
    return(data.table::fread(phenotype_file))
}
memPHENO <- memoise::memoise(get_pheno_df)

get_caudate_eqtls <- function(){
    mashr_file = paste0("../../../mashr/summary_table/_m/",
                        "Brainseq_LIBD_caudate_specific_4features.eGenes.txt.gz")
    return(data.table::fread(mashr_file) %>%
           filter(Type == feature_map(feature)) %>%
           select(gene_id, variant_id))
}
memCAUDATE <- memoise::memoise(get_caudate_eqtls)

get_eqtl_df <- function(){
    fastqtl_file = paste0("/ceph/projects/v4_phase3_paper/analysis/
 ↪eqtl_analysis/all/", feature,
                         "/expression_gct/prepare_expression/
 ↪fastqtl_permutation/_m/",
                         "Brainseq_LIBD.genes.txt.gz")
    eqtl_df = data.table::fread(fastqtl_file) %>%
        filter(gene_id %in% memCAUDATE()$gene_id) %>%
        arrange(qval)
    return(eqtl_df)
}
memEQTL <- memoise::memoise(get_eqtl_df)

get_genotypes <- function(){
    traw_file = paste0("/ceph/projects/brainseq/genotype/download/topmed/
 ↪convert2plink/",
                      "filter_maf_01/a_transpose/_m/LIBD_Brain_TopMed.traw")
    traw = data.table::fread(traw_file) %>% rename_with(~ gsub('\\_.*', '', .x))
    return(traw)
}
memSNPs <- memoise::memoise(get_genotypes)
```

1.1.2 Simple functions

[4]:

```
feature_map <- function(feature){
    return(list("genes"="Gene", "transcripts"= "Transcript",
                "exons"= "Exon", "junctions"= "Junction")[[feature]])
}

get_geno_annot <- function(){
    return(memSNPs() %>% select(CHR, SNP, POS, COUNTED, ALT))
}

get_snps_df <- function(){
    return(memSNPs() %>% select("SNP", starts_with("Br")))
}

letter_snp <- function(number, a0, a1){
    if(is.na(number)){ return(NA) }
    if( length(a0) == 1 & length(a1) == 1){
        seps = ""; collapse=""
    } else {
        seps = " "; collapse=NULL
    }
    return(paste(paste0(rep(a0, number), collapse = collapse),
                 paste0(rep(a1, (2-number)), collapse = collapse), sep=seps))
}

get_snp_df <- function(variant_id, gene_id){
    zz = get_geno_annot() %>% filter(SNP == variant_id)
    xx = get_snps_df() %>% filter(SNP == variant_id) %>%
        column_to_rownames("SNP") %>% t %>% as.data.frame %>%
        rownames_to_column("BrNum") %>% mutate(COUNTED=zz$COUNTED, ALT=zz$ALT)␣
 ↪%>%
        rename("SNP"=all_of(variant_id))
    yy = memRES()[gene_id, ] %>% t %>% as.data.frame %>%
        rownames_to_column("RNum") %>% inner_join(memPHENO(), by="RNum")
    ## Annotated SNPs
    letters = c()
    for(ii in seq_along(xx$COUNTED)){
        a0 = xx$COUNTED[ii]; a1 = xx$ALT[ii]; number = xx$SNP[ii]
        letters <- append(letters, letter_snp(number, a0, a1))
    }
    xx = xx %>% mutate(LETTER=letters, ID=paste(SNP, LETTER, sep="\n"))
    df = inner_join(xx, yy, by="BrNum") %>% mutate_if(is.character, as.factor)
    return(df)
}
memDF <- memoise::memoise(get_snp_df)

save_ggplots <- function(fn, p, w, h){
    for(ext in c('.pdf', '.png', '.svg')){
        ggsave(paste0(fn, ext), plot=p, width=w, height=h)
    }
}

get_gene_symbol <- function(gene_id){
    ensemblID = gsub("\\..*", "", gene_id)
    geneid = memMART() %>% filter(ensembl_gene_id == gsub("\\..*", "", gene_id))
    if(dim(geneid)[1] == 0){
        return("")
    } else {
        return(geneid$external_gene_name)
    }
}

plot_simple_eqtl <- function(fn, gene_id, variant_id, eqtl_annot){
    bxp = memDF(variant_id, gene_id) %>%
        ggboxplot(x="ID", y=gene_id, fill="Region", color="Region",␣
 ↪add="jitter",
                  xlab=variant_id, ylab="Residualized Expression", outlier.
↪shape=NA,
                  add.params=list(alpha=0.5), alpha=0.4, legend="bottom",
                  palette="npg", ggtheme=theme_pubr(base_size=20, border=TRUE))␣
↪+
        font("xy.title", face="bold") +
        ggtitle(paste(get_gene_symbol(gene_id), gene_id, eqtl_annot, sep='\n'))␣
↪+
        theme(plot.title = element_text(hjust = 0.5, face="bold"))
    print(bxp)
    save_ggplots(fn, bxp, 7, 7)
}
```

1.1.3 GWAS plots

[5]:

```
get_gwas_snps <- function(){
    gwas_snp_file = paste0('/ceph/projects/v4_phase3_paper/inputs/sz_gwas/
↪pgc2_clozuk/',
                           'map_phase3/_m/libd_hg38_pgc2sz_snps_p5e_minus8.tsv')
    gwas_df = data.table::fread(gwas_snp_file) %>% arrange(P)
    return(gwas_df)
}
memGWAS <- memoise::memoise(get_gwas_snps)

get_gwas_snp <- function(variant){
    return(memGWAS() %>% filter(our_snp_id == variant))
}

get_risk_allele <- function(variant){
    gwas_snp = get_gwas_snp(variant)
    if(gwas_snp$OR > 1){
        ra = gwas_snp$A1
    }else{
        ra = gwas_snp$A2
    }
    return(ra)
}

get_eqtl_gwas_df <- function(){
    return(memEQTL() %>% inner_join(memGWAS(), by=c("variant_id"="our_snp_id")))
}

get_gwas_ordered_snp_df <- function(variant_id, gene_id,␣
↪pgc2_a1_same_as_our_counted, OR){
    df = memDF(variant_id, gene_id)
    if(!pgc2_a1_same_as_our_counted){ # Fix bug with matching alleles!
```

```
        if(OR < 1){ df = df %>% mutate(SNP = 2-SNP, ID=paste(SNP, LETTER,␣
 ↪sep="\n")) }
    } else {
        if(OR > 1){ df = df %>% mutate(SNP = 2-SNP, ID=paste(SNP, LETTER,␣
 ↪sep="\n")) }
    }
    return(df)
}

plot_gwas_eqtl <- function(fn, gene_id, variant_id, eqtl_annot,
                           pgc2_a1_same_as_our_counted, OR, title){
    dt = get_gwas_ordered_snp_df(variant_id, gene_id,␣
 ↪pgc2_a1_same_as_our_counted, OR)
    y0 = quantile(dt[[gene_id]], probs=c(0.05))[[1]] - 0.26
    y1 = quantile(dt[[gene_id]], probs=c(0.95))[[1]] + 0.26
    bxp = dt %>% mutate_if(is.character, as.factor) %>%
        ggboxplot(x="ID", y=gene_id, fill="Region", color="Region",␣
 ↪add="jitter",
                  xlab=variant_id, ylab="Residualized Expression", outlier.
 ↪shape=NA,
                  add.params=list(alpha=0.5), alpha=0.4, legend="bottom",␣
 ↪lims=c(y0,y1),
                  palette="npg", ggtheme=theme_pubr(base_size=20, border=TRUE))␣
 ↪+
        font("xy.title", face="bold") + ggtitle(title) +
        theme(plot.title = element_text(hjust = 0.5, face="bold"))
    print(bxp)
    save_ggplots(fn, bxp, 7, 8)
}
```

1.2 Plot eQTL

[6]:

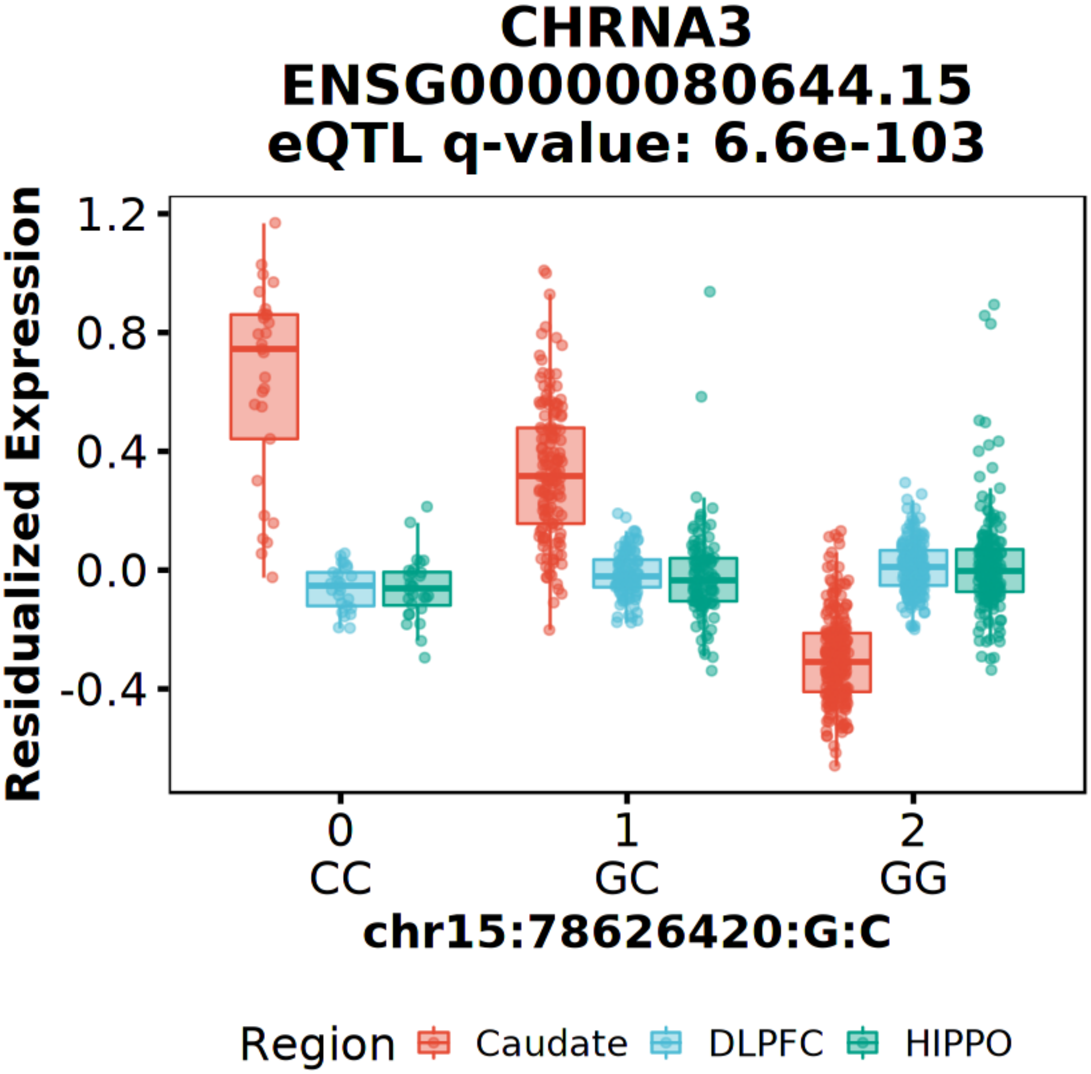
```
eqtl_df = memEQTL()
eqtl_df %>% head(5)
```

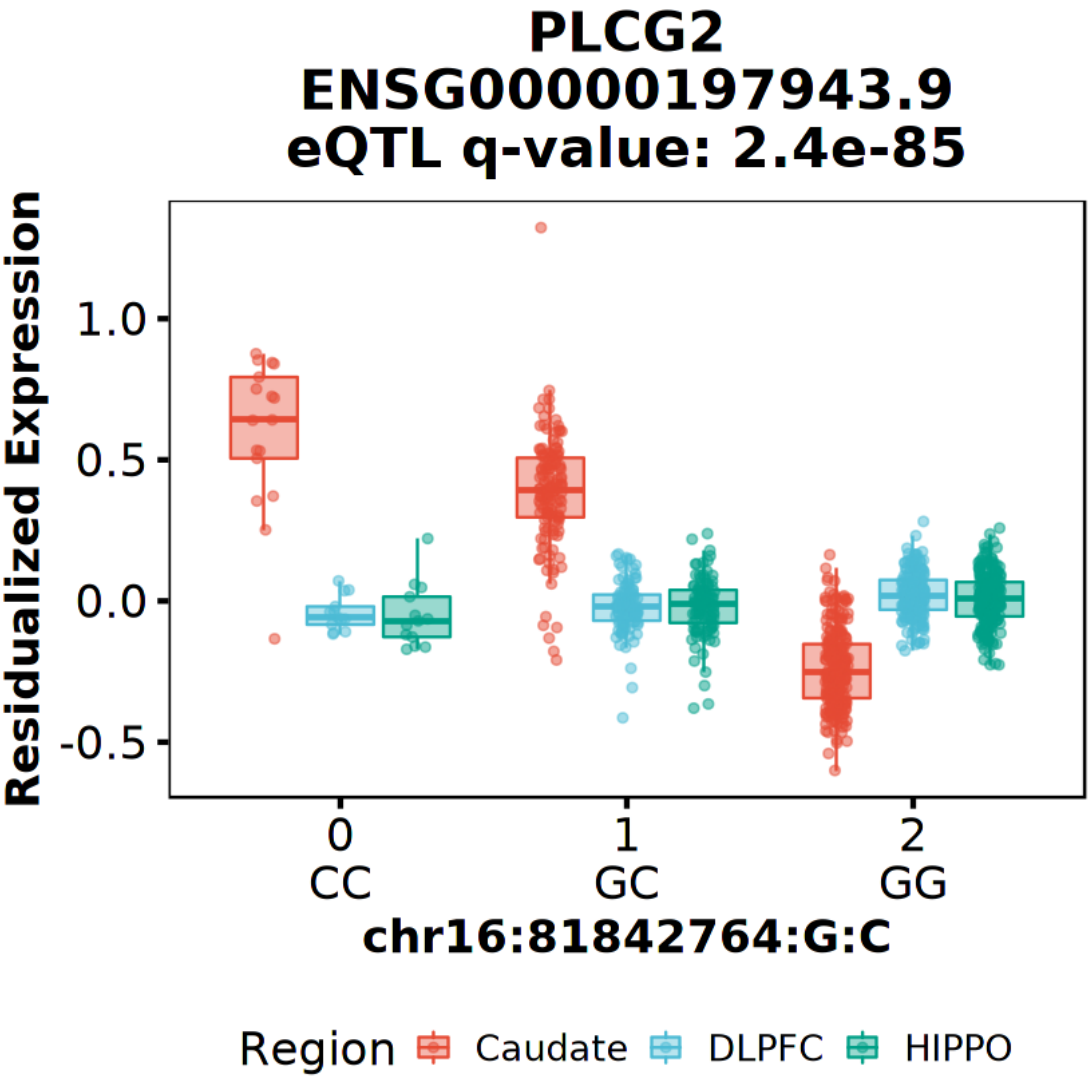
A data.table: 5 × 19

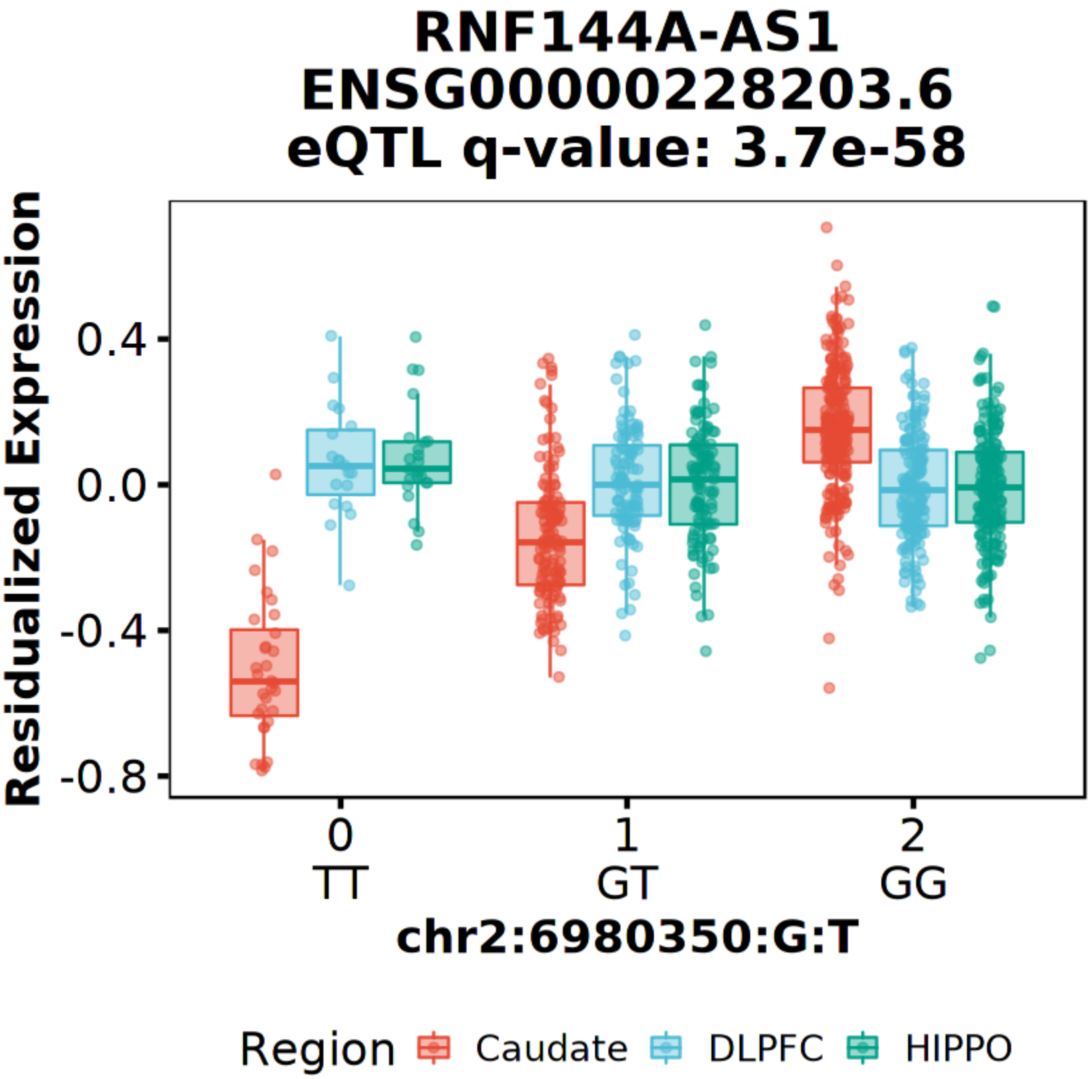
gene_id <chr>	num_var <int>	beta_shape1 <dbl>	beta_shape2 <dbl>	true_df <dbl>	pval_true_df <dbl>
ENSG00000080644.15	4093	1.02864	537.302	356.264	6.90647e-105
ENSG00000197943.9	5911	1.04611	1037.640	361.232	2.33294e-86
ENSG00000228203.6	4228	1.04032	649.263	357.933	5.64801e-60
ENSG00000037280.15	5072	1.03289	736.139	356.145	5.02450e-60
ENSG00000113494.16	4082	1.05987	406.859	345.553	7.70764e-54

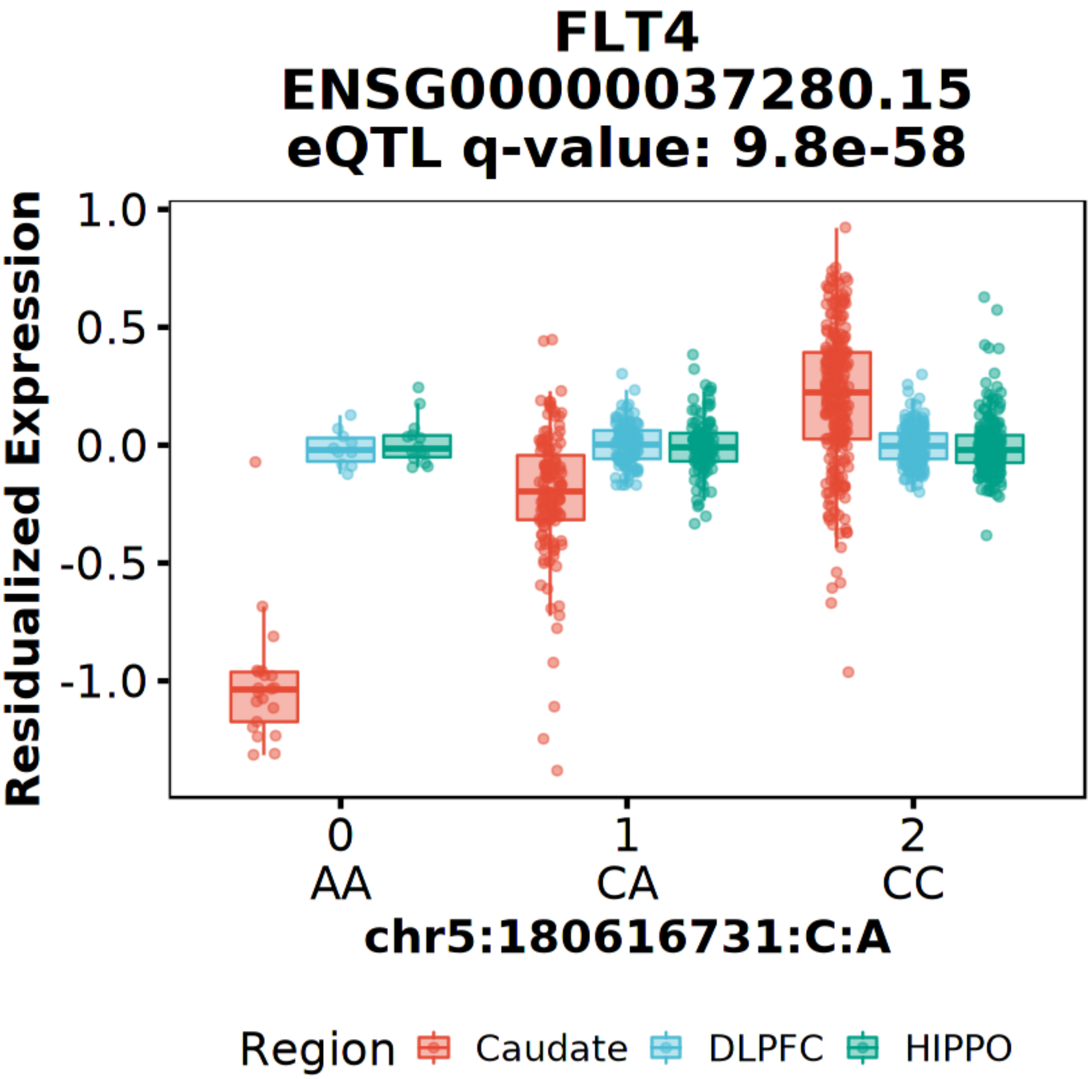
1.2.1 Top 5 eQTLs

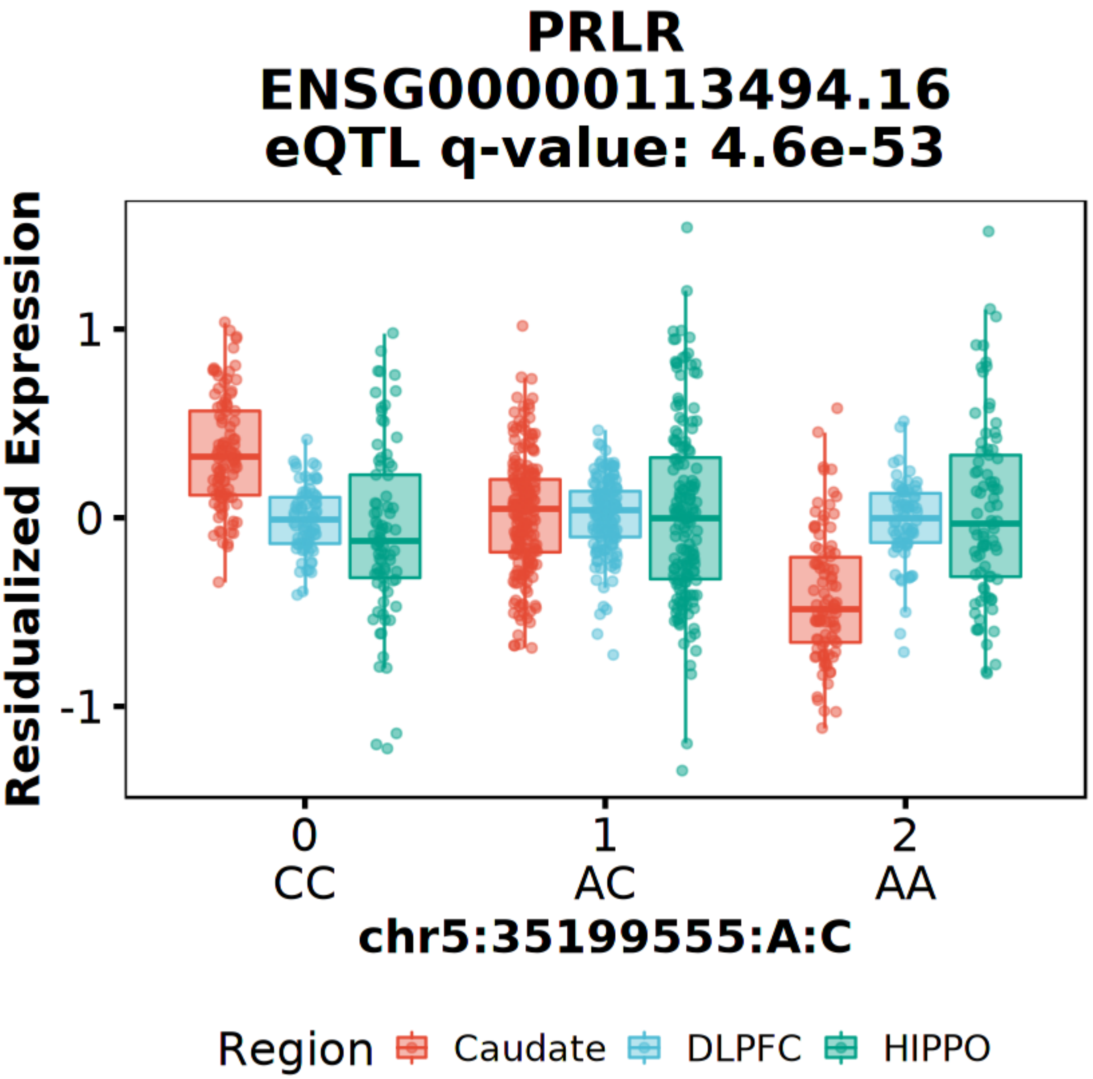
[7]:
```
for(num in 1:5){
    variant_id = memEQTL()$variant_id[num]
    gene_id = memEQTL()$gene_id[num]
    eqtl_annot = paste("eQTL q-value:", signif(memEQTL()$qval[num], 2))
    fn = paste0("top_",num,"_interacting_eqtl")
    plot_simple_eqtl(fn, gene_id, variant_id, eqtl_annot)
}
```




PLCG2
ENSG00000197943.9
eQTL q-value: 2.4e-85
Residualized Expression
1.0
0.5
0.0
-0.5
0
CC
1
GC
2
GG
chr16:81842764:G:C
Region
Caudate
DLPFC
HIPPO


RNF144A-AS1
ENSG00000228203.6
eQTL q-value: 3.7e-58
Residualized Expression
0.4
0.0
-0.4
-0.8
0
TT
1
GT
2
GG
chr2:6980350:G:T
Region
Caudate
DLPFC
HIPPO


FLT4
ENSG00000037280.15
eQTL q-value: 9.8e-58
Residualized Expression
1.0
0.5
0.0
-0.5
-1.0
0
AA
1
CA
2
CC
chr5:180616731:C:A
Region
Caudate
DLPFC
HIPPO



1.2.2 Top 5 GWAS associated eQTLs

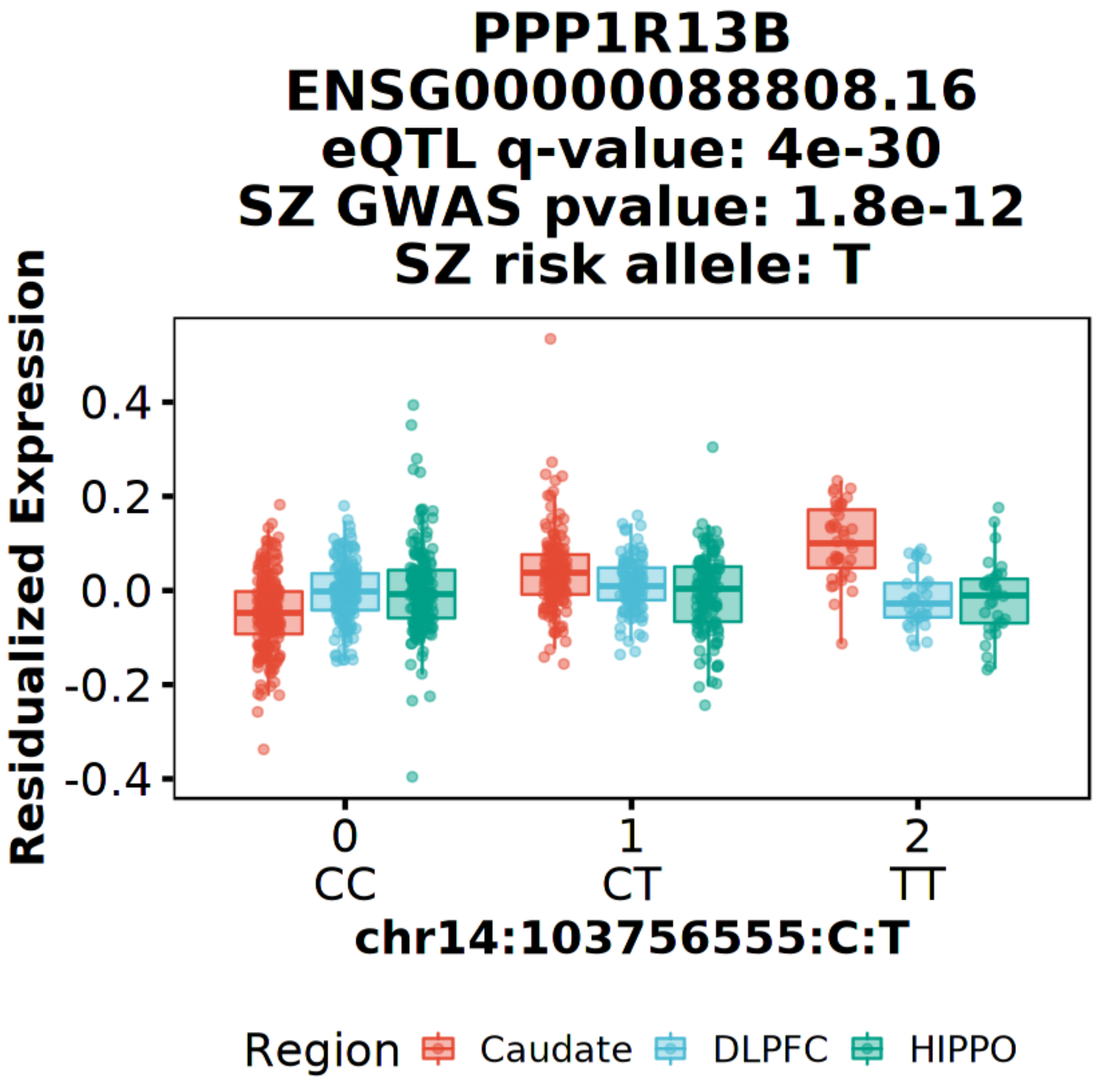
```
[8]: eqtl_gwas_df = get_eqtl_gwas_df()
     eqtl_gwas_df %>% head(5)
```

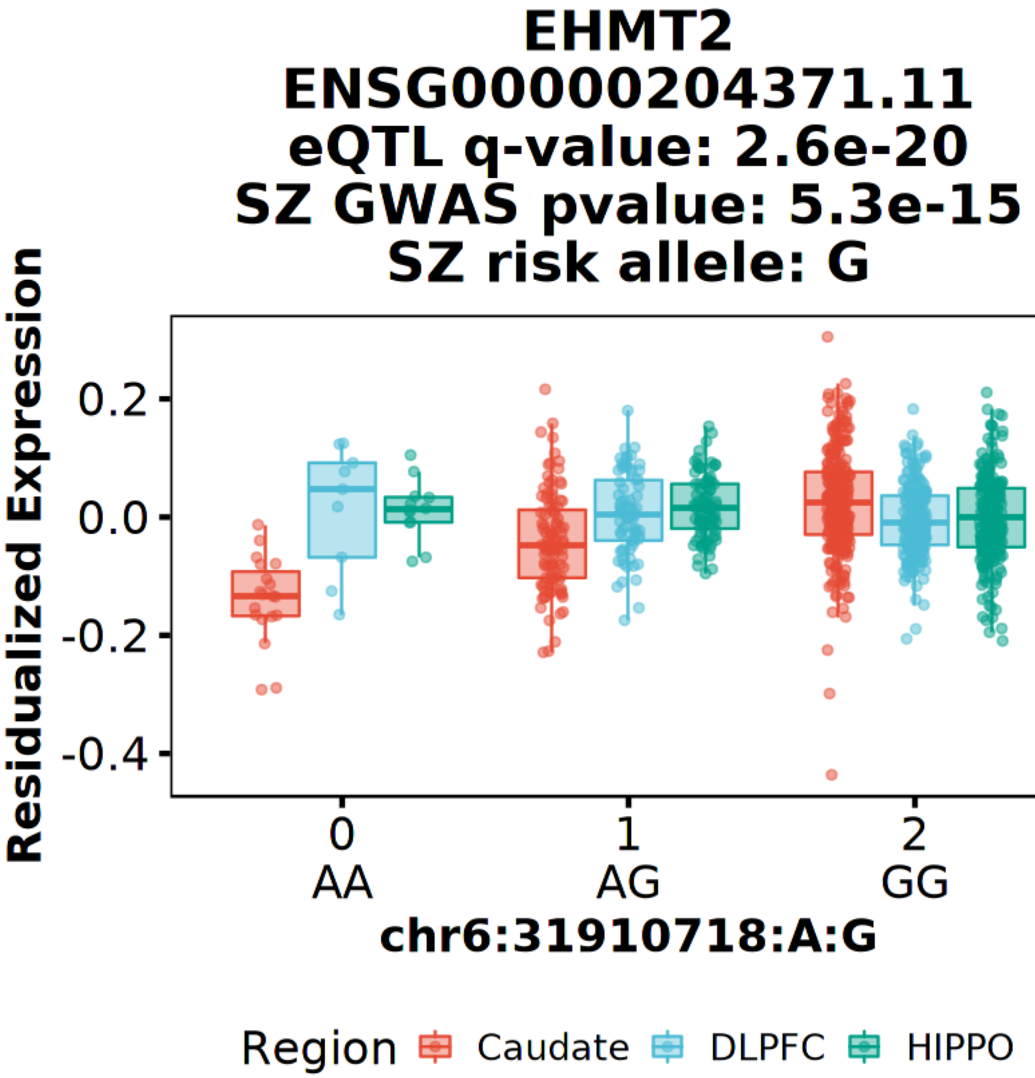
A data.table: 5 × 41

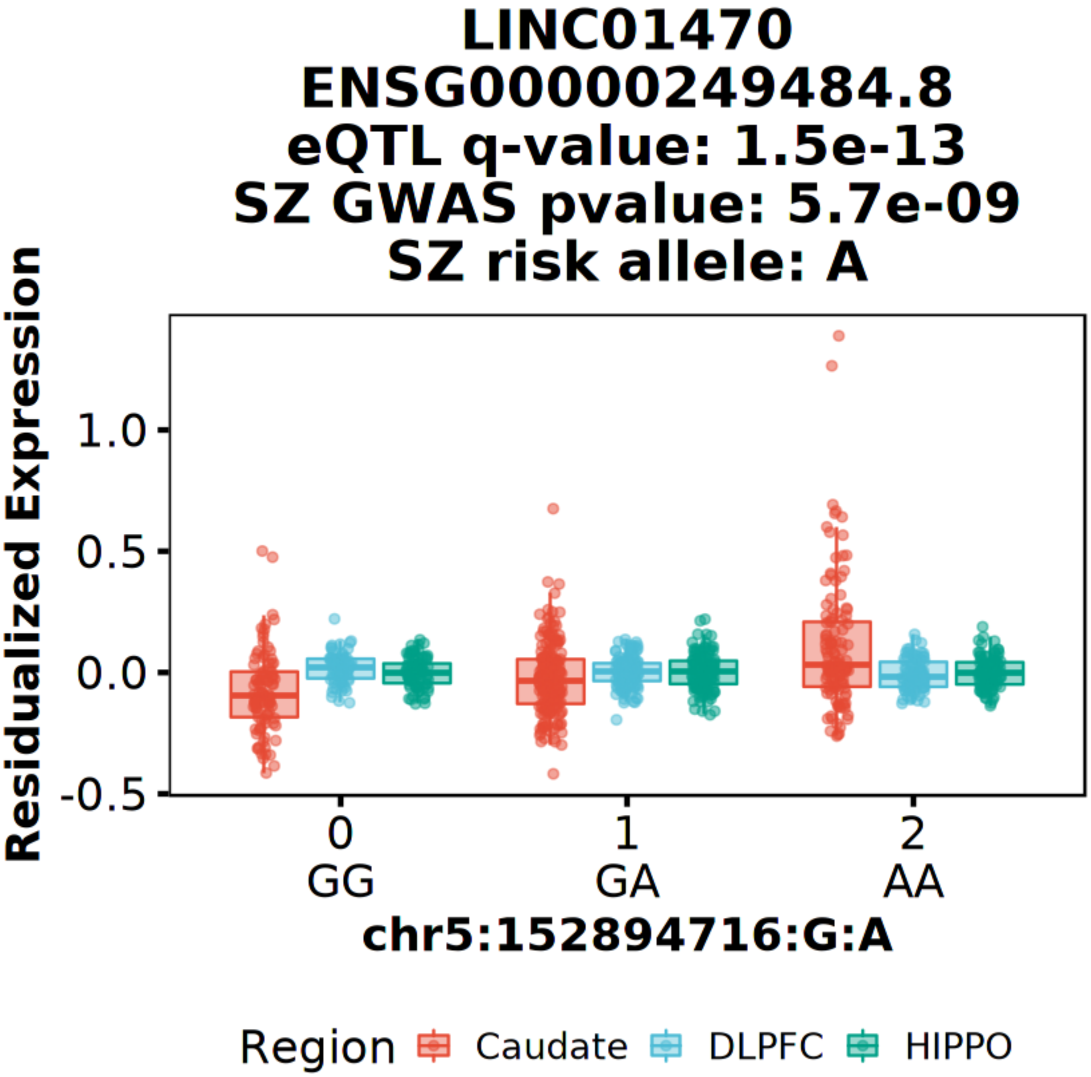
gene_id	num_var	beta_shape1	beta_shape2	true_df	pval_true_df
<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>
ENSG00000088808.16	4249	1.06500	394.626	344.221	1.31453e-31
ENSG00000204371.11	5739	1.06250	242.393	324.467	5.55941e-22
ENSG00000249484.8	4168	1.02535	326.601	351.574	5.52907e-16
ENSG00000149930.17	1637	1.03636	253.345	354.893	1.84336e-08
ENSG00000174938.14	1592	1.06550	228.280	346.685	2.12798e-06

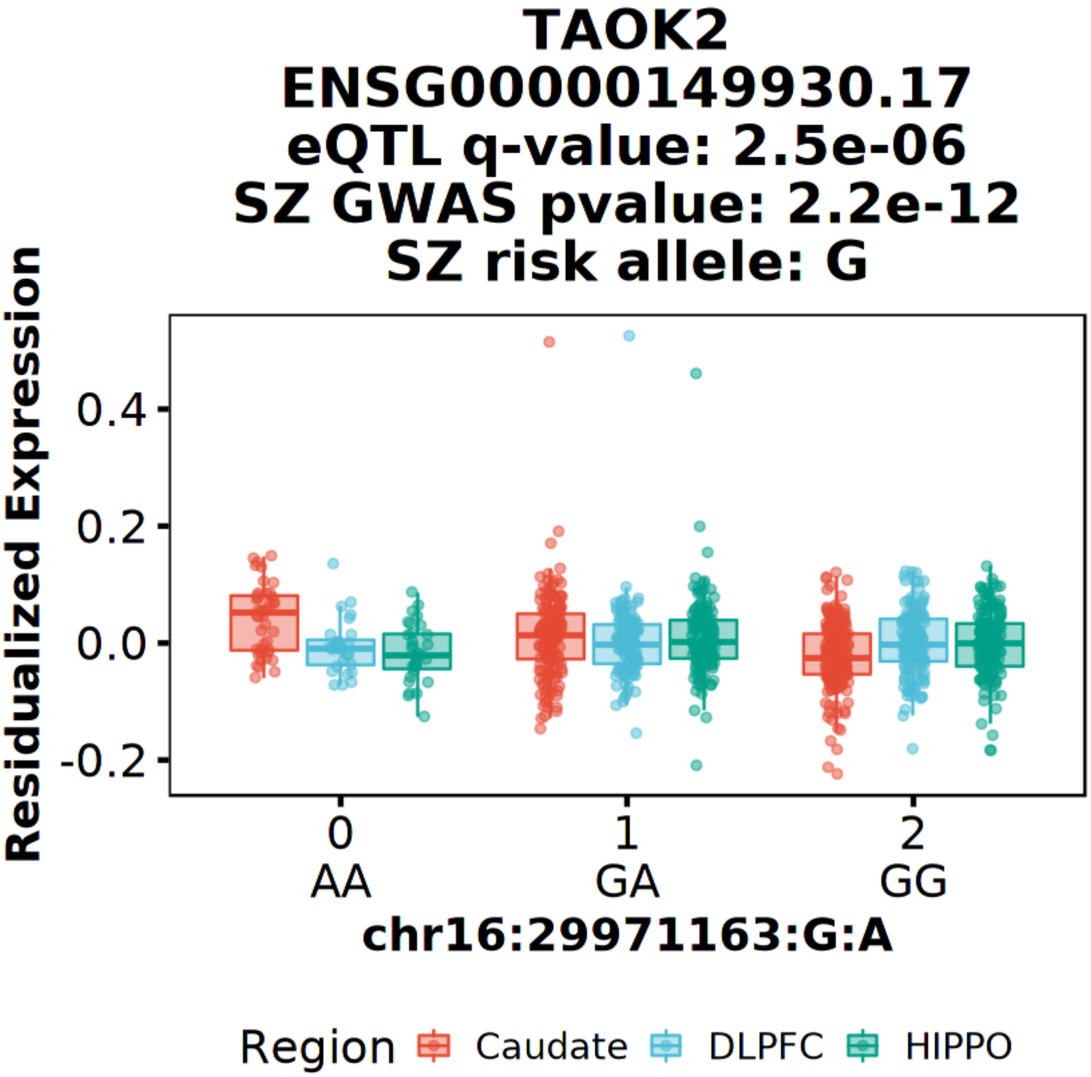
[9]:

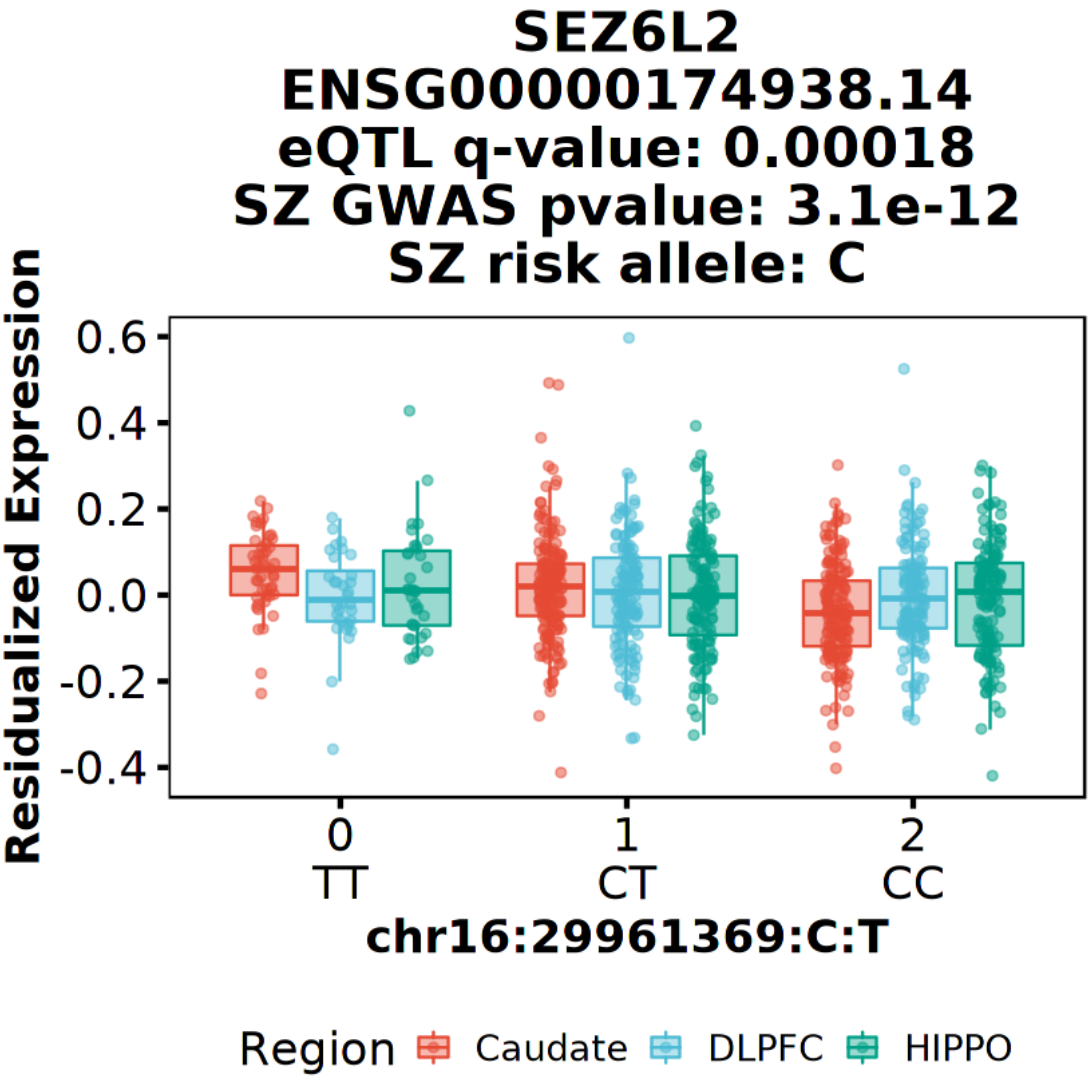
```r
for(num in 1:5){
    fn = paste("top",num,"interacting_eqtl_in_gwas_significant_snps", sep="_")
    variant_id = eqtl_gwas_df$variant_id[num]
    gene_id = eqtl_gwas_df$gene_id[num]
    pgc2_a1_same_as_our_counted = eqtl_gwas_df$pgc2_a1_same_as_our_counted[num]
    OR = eqtl_gwas_df$OR[num]
    eqtl_annot = paste("eQTL q-value:", signif(eqtl_gwas_df$qval[num], 2))
    gwas_annot = paste("SZ GWAS pvalue:", signif(eqtl_gwas_df$P[num], 2))
    risk_annot = paste("SZ risk allele:",␣
 ↪get_risk_allele(eqtl_gwas_df$variant_id[num]))
    title = paste(get_gene_symbol(gene_id), gene_id, eqtl_annot,
                  gwas_annot, risk_annot, sep='\n')
    plot_gwas_eqtl(fn, gene_id, variant_id, eqtl_annot,
                   pgc2_a1_same_as_our_counted, OR, title)
}
```




EHMT2
ENSG00000204371.11
eQTL q-value: 2.6e-20
SZ GWAS pvalue: 5.3e-15
SZ risk allele: G
Residualized Expression
0.2
0.0
-0.2
-0.4
0
AA
1
AG
2
GG
chr6:31910718:A:G
Region
Caudate
DLPFC
HIPPO


LINC01470
ENSG00000249484.8
eQTL q-value: 1.5e-13
SZ GWAS pvalue: 5.7e-09
SZ risk allele: A
Residualized Expression
1.0
0.5
0.0
-0.5
0
GG
1
GA
2
AA
chr5:152894716:G:A
Region
Caudate
DLPFC
HIPPO


TAOK2
ENSG00000149930.17
eQTL q-value: 2.5e-06
SZ GWAS pvalue: 2.2e-12
SZ risk allele: G
Residualized Expression
0.4
0.2
0.0
-0.2
0
AA
1
GA
2
GG
chr16:29971163:G:A
Region
Caudate
DLPFC
HIPPO



1.3 Session Info

```
[10]: Sys.time()
      proc.time()
      options(width = 120)
      sessioninfo::session_info()
```

```
[1] "2021-09-08 16:13:10 EDT"
```

```
    user   system  elapsed
4536.706  765.740  727.886
```

```
 Session info
setting  value
```

```
version  R version 4.0.3 (2020-10-10)
os       Arch Linux
system   x86_64, linux-gnu
ui       X11
language (EN)
collate  en_US.UTF-8
ctype    en_US.UTF-8
tz       America/New_York
date     2021-09-08

 Packages
package     * version  date       lib source
abind         1.4-5    2016-07-21 [1] CRAN (R 4.0.2)
assertthat    0.2.1    2019-03-21 [1] CRAN (R 4.0.2)
backports     1.2.1    2020-12-09 [1] CRAN (R 4.0.2)
base64enc     0.1-3    2015-07-28 [1] CRAN (R 4.0.2)
broom         0.7.9    2021-07-27 [1] CRAN (R 4.0.3)
cachem        1.0.6    2021-08-19 [1] CRAN (R 4.0.3)
Cairo         1.5-12.2 2020-07-07 [1] CRAN (R 4.0.2)
car           3.0-11   2021-06-27 [1] CRAN (R 4.0.3)
carData       3.0-4    2020-05-22 [1] CRAN (R 4.0.2)
cellranger    1.1.0    2016-07-27 [1] CRAN (R 4.0.2)
cli           3.0.1    2021-07-17 [1] CRAN (R 4.0.3)
colorspace    2.0-2    2021-06-24 [1] CRAN (R 4.0.3)
crayon        1.4.1    2021-02-08 [1] CRAN (R 4.0.3)
curl          4.3.2    2021-06-23 [1] CRAN (R 4.0.3)
data.table    1.14.0   2021-02-21 [1] CRAN (R 4.0.3)
DBI           1.1.1    2021-01-15 [1] CRAN (R 4.0.2)
dbplyr        2.1.1    2021-04-06 [1] CRAN (R 4.0.3)
digest        0.6.27   2020-10-24 [1] CRAN (R 4.0.2)
dplyr       * 1.0.7    2021-06-18 [1] CRAN (R 4.0.3)
ellipsis      0.3.2    2021-04-29 [1] CRAN (R 4.0.3)
evaluate      0.14     2019-05-28 [1] CRAN (R 4.0.2)
fansi         0.5.0    2021-05-25 [1] CRAN (R 4.0.3)
farver        2.1.0    2021-02-28 [1] CRAN (R 4.0.3)
fastmap       1.1.0    2021-01-25 [1] CRAN (R 4.0.2)
forcats     * 0.5.1    2021-01-27 [1] CRAN (R 4.0.2)
foreign       0.8-80   2020-05-24 [2] CRAN (R 4.0.3)
fs            1.5.0    2020-07-31 [1] CRAN (R 4.0.2)
generics      0.1.0    2020-10-31 [1] CRAN (R 4.0.2)
ggplot2     * 3.3.5    2021-06-25 [1] CRAN (R 4.0.3)
ggpubr      * 0.4.0    2020-06-27 [1] CRAN (R 4.0.2)
ggsci         2.9      2018-05-14 [1] CRAN (R 4.0.2)
ggsignif      0.6.2    2021-06-14 [1] CRAN (R 4.0.3)
glue          1.4.2    2020-08-27 [1] CRAN (R 4.0.2)
gtable        0.3.0    2019-03-25 [1] CRAN (R 4.0.2)
haven         2.4.3    2021-08-04 [1] CRAN (R 4.0.3)
hms           1.1.0    2021-05-17 [1] CRAN (R 4.0.3)
```

```
htmltools     0.5.2    2021-08-25 [1] CRAN (R 4.0.3)
httr          1.4.2    2020-07-20 [1] CRAN (R 4.0.2)
IRdisplay     1.0      2021-01-20 [1] CRAN (R 4.0.2)
IRkernel      1.2      2021-05-11 [1] CRAN (R 4.0.3)
jsonlite      1.7.2    2020-12-09 [1] CRAN (R 4.0.2)
labeling      0.4.2    2020-10-20 [1] CRAN (R 4.0.2)
lifecycle     1.0.0    2021-02-15 [1] CRAN (R 4.0.3)
lubridate     1.7.10   2021-02-26 [1] CRAN (R 4.0.3)
magrittr      2.0.1    2020-11-17 [1] CRAN (R 4.0.2)
memoise       2.0.0    2021-01-26 [1] CRAN (R 4.0.2)
modelr        0.1.8    2020-05-19 [1] CRAN (R 4.0.2)
munsell       0.5.0    2018-06-12 [1] CRAN (R 4.0.2)
openxlsx      4.2.4    2021-06-16 [1] CRAN (R 4.0.3)
pbdZMQ        0.3-5    2021-02-10 [1] CRAN (R 4.0.3)
pillar        1.6.2    2021-07-29 [1] CRAN (R 4.0.3)
pkgconfig     2.0.3    2019-09-22 [1] CRAN (R 4.0.2)
purrr       * 0.3.4    2020-04-17 [1] CRAN (R 4.0.2)
R.methodsS3   1.8.1    2020-08-26 [1] CRAN (R 4.0.3)
R.oo          1.24.0   2020-08-26 [1] CRAN (R 4.0.3)
R.utils       2.10.1   2020-08-26 [1] CRAN (R 4.0.3)
R6            2.5.1    2021-08-19 [1] CRAN (R 4.0.3)
Rcpp          1.0.7    2021-07-07 [1] CRAN (R 4.0.3)
readr       * 2.0.1    2021-08-10 [1] CRAN (R 4.0.3)
readxl        1.3.1    2019-03-13 [1] CRAN (R 4.0.2)
repr          1.1.3    2021-01-21 [1] CRAN (R 4.0.2)
reprex        2.0.1    2021-08-05 [1] CRAN (R 4.0.3)
rio           0.5.27   2021-06-21 [1] CRAN (R 4.0.3)
rlang         0.4.11   2021-04-30 [1] CRAN (R 4.0.3)
rstatix       0.7.0    2021-02-13 [1] CRAN (R 4.0.3)
rstudioapi    0.13     2020-11-12 [1] CRAN (R 4.0.2)
rvest         1.0.1    2021-07-26 [1] CRAN (R 4.0.3)
scales        1.1.1    2020-05-11 [1] CRAN (R 4.0.2)
sessioninfo   1.1.1    2018-11-05 [1] CRAN (R 4.0.2)
stringi       1.7.4    2021-08-25 [1] CRAN (R 4.0.3)
stringr     * 1.4.0    2019-02-10 [1] CRAN (R 4.0.2)
svglite       2.0.0    2021-02-20 [1] CRAN (R 4.0.3)
systemfonts   1.0.2    2021-05-11 [1] CRAN (R 4.0.3)
tibble      * 3.1.4    2021-08-25 [1] CRAN (R 4.0.3)
tidyr       * 1.1.3    2021-03-03 [1] CRAN (R 4.0.3)
tidyselect    1.1.1    2021-04-30 [1] CRAN (R 4.0.3)
tidyverse   * 1.3.1    2021-04-15 [1] CRAN (R 4.0.3)
tzdb          0.1.2    2021-07-20 [1] CRAN (R 4.0.3)
utf8          1.2.2    2021-07-24 [1] CRAN (R 4.0.3)
uuid          0.1-4    2020-02-26 [1] CRAN (R 4.0.2)
vctrs         0.3.8    2021-04-29 [1] CRAN (R 4.0.3)
withr         2.4.2    2021-04-18 [1] CRAN (R 4.0.3)
xml2          1.3.2    2020-04-23 [1] CRAN (R 4.0.2)
zip           2.2.0    2021-05-31 [1] CRAN (R 4.0.3)
```

```
[1] /home/jbenja13/R/x86_64-pc-linux-gnu-library/4.0
[2] /usr/lib/R/library
```