

# main\_transcripts

September 5, 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 = "transcripts"
```

#### 1.1.1 Cached functions

```
[3]: get_eqtl_df <- function(){  
      eGenes_file = paste0('/ceph/projects/v4_phase3_paper/analysis/eqtl_analysis/  
      ↪all/',  
                           feature, '/expression_gct/prepare_expression/  
      ↪fastqtl_permutation/',  
                           '_m/Brainseq_LIBD.genes.txt.gz')  
      eGenes = data.table::fread(eGenes_file) %>%  
        select(gene_id, variant_id, maf, slope, slope_se, pval_nominal, qval) ↪  
      ↪%>%  
        arrange(qval)  
      return(eGenes)  
    }  
    memEQTL <- memoise::memoise(get_eqtl_df)  
  
    get_residualized_df <- function(){  
      expr_file = paste0("/ceph/projects/v4_phase3_paper/analysis/eqtl_analysis/  
      ↪all/",  
                          feature, "/expression_gct/covariates/  
      ↪residualized_expression/_m/",  
                          feature, "_residualized_expression.csv")  
      return(data.table::fread(expr_file) %>% column_to_rownames("gene_id"))  
    }  
    memRES <- memoise::memoise(get_residualized_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("BrNum")
  ## 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_biomart_df <- function(){
  biomart = data.table::fread("../_h/biomart.csv")
}
memMART <- memoise::memoise(get_biomart_df)

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, prefix){
  bxp = memDF(variant_id, gene_id) %>%
    ggboxplot(x="ID", y=gene_id, fill="red", add="jitter", xlab="",
              ylab="Residualized Expression", outlier.shape=NA,
              add.params=list(alpha=0.5), alpha=0.4,
              ggtheme=theme_pubr(base_size=20, border=TRUE)) +
    font("xy.title", face="bold") +
    ggtitle(paste(prefix, 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('.././summary_table/_m/Brainseq_LIBD_caudate',
    '_4features_PGC2.signifpairs.txt.gz')
  gwas_df = data.table::fread(gwas_snp_file) %>% filter(Type == "A1",
    ↪feature_map(feature)) %>%
    select(c("variant_id", "gene_id", "rsid", "hg38chr", "OR", "SE", "P", "A1",
    ↪"A2", "pgc2_a1_same_as_our_counted", "is_index_snp")) %>%
    distinct() %>% arrange(P)
  return(gwas_df)
}
memGWAS <- memoise::memoise(get_gwas_snps)

get_gwas_snp <- function(variant){
  return(memGWAS() %>% filter(variant_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="variant_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){
    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){
  bxp = get_gwas_ordered_snp_df(variant_id, gene_id,
                                pgc2_a1_same_as_our_counted, OR) %>%
    mutate_if(is.character, as.factor) %>%
    ggboxplot(x="ID", y=gene_id, fill="red", add="jitter", xlab="",
              ylab="Residualized Expression", outlier.shape=NA,
              add.params=list(alpha=0.5), alpha=0.4,
              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, 7)
}

```

## 1.2 Plot eQTL

### 1.2.1 DRD2 plot

```

[6]: drd2_short = "ENST00000346454.7"; drd2_long = "ENST00000362072.7"
drd2_df = memEQTL() %>% filter(gene_id %in% c(drd2_short, drd2_long))
drd2_df

```

A data.table: 2 × 7

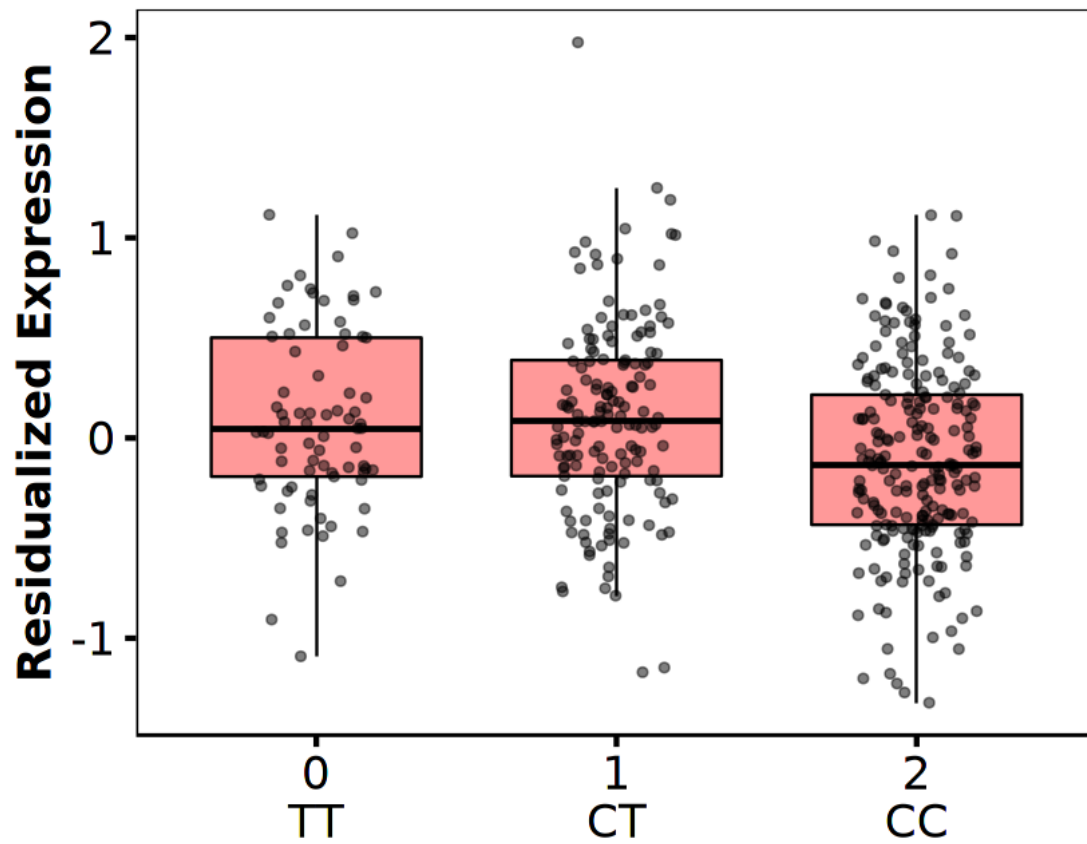
	gene_id <chr>	variant_id <chr>	maf <dbl>	slope <dbl>	slope_se <dbl>	pval_nomi <dbl>
	ENST00000346454.7	chr11:113371155:C:T	0.341986	0.172006	0.0375699	6.22498e-0
	ENST00000362072.7	chr11:113637703:A:G	0.436795	-0.121449	0.0363983	9.25246e-0

```

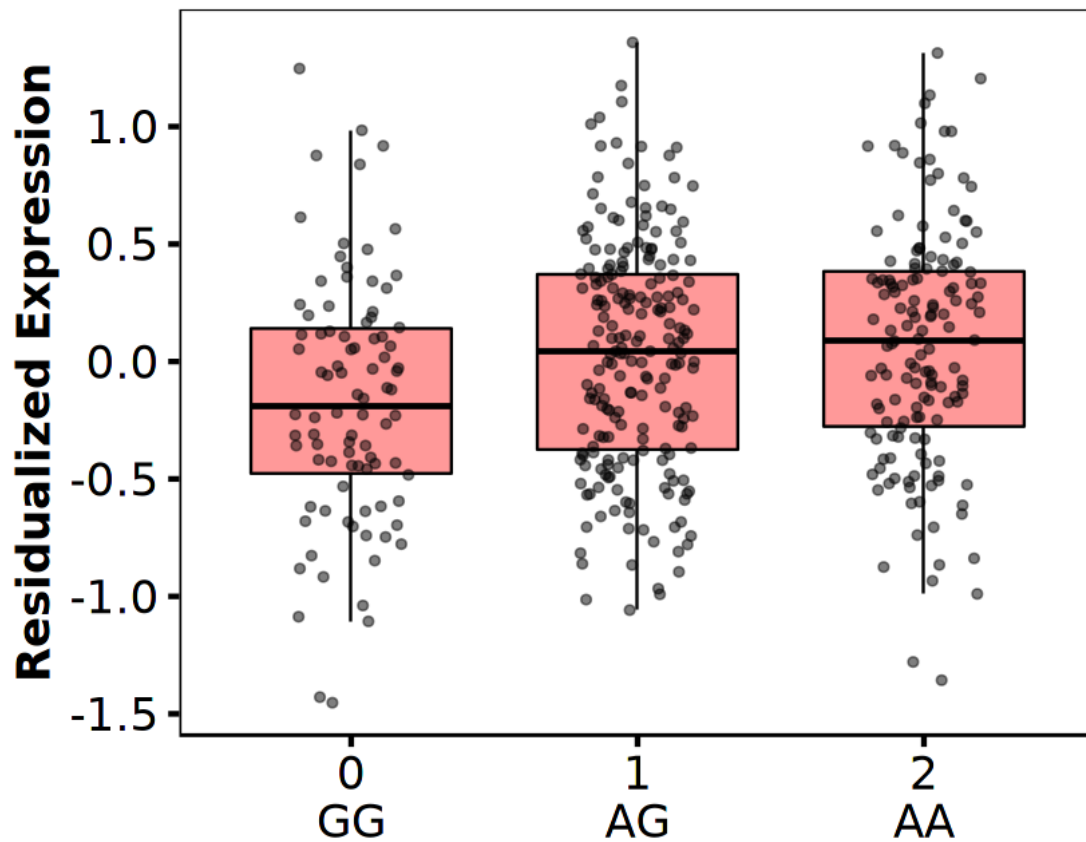
[7]: drd2_map = list("ENST00000346454.7"="D2S", "ENST00000362072.7"="D2L")
drd2_map_long = list("ENST00000346454.7"= "DRD2 Short", "ENST00000362072.7"=
  ↪ "DRD2 Long")
for(x in seq_along(drd2_df$gene_id)){
  fn = paste("drd2_eqtl", drd2_map[[drd2_df$gene_id[x]]], sep="_")
  eqtl_annot = paste("eQTL q-value:", signif(drd2_df$qval[x], 2))
  prefix = drd2_map_long[[drd2_df$gene_id[x]]]
  plot_simple_eqtl(fn, drd2_df$gene_id[x], drd2_df$variant_id[x], eqtl_annot,
  ↪ prefix)
}

```

**DRD2 Short**  
**ENST00000346454.7**  
**eQTL q-value: 0.022**



## DRD2 Long ENST00000362072.7 eQTL q-value: 0.39



### 1.2.2 GWAS association

```
[8]: memGWAS() %>% filter(gene_id %in% c(drd2_short, drd2_long))
```

A data.table: 0 × 11

variant_id	gene_id	rsid	hg38chr	OR	SE	P	A1	A2	p
<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<chr>	<chr>	<

### 1.3 Session Info

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

```
[1] "2021-09-05 15:48:20 EDT"
```

```
      user system elapsed
898.425 252.393 177.310
```

#### 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     2021-09-05
```

#### 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)



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