

# main

March 4, 2022

## 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 Basic loading functions

```
[3]: get_residualized_df <- function(){  
      expr_file = paste0("/ceph/projects/v4_phase3_paper/analysis/eql_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_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_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 eQTL and helpful functions

```

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

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_caudeate_eqtls <- function(){
    mashr_file = paste0("../.../meta_analysis/summary_table/_m/",
                       "BrainSeq_caudeate_eQTL.txt.gz")
    return(data.table::fread(mashr_file) %>%
           filter(Type == feature_map(feature)) %>%
           select(gene_id, variant_id, AA, EA))
}
memCAUDate <- memoise::memoise(get_caudeate_eqtls)

```

### 1.1.3 Basic eQTL plotting functions

```

[5]: get_genotype_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_genotype_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") %>% inner_join(memPHENO(), by="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)

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)
  }
}
}

```

```

[6]: plot_simple_eqtl <- function(fn, gene_id, variant_id, eqtl_annot, y0=NULL,
  ↪ y1=NULL){
  if(is.null(y0)){ y0 = quantile(memDF(variant_id, gene_id)[[gene_id]],
  ↪ probs=c(0.01))[[1]] - 0.2 }
  if(is.null(y1)){ y1 = quantile(memDF(variant_id, gene_id)[[gene_id]],
  ↪ probs=c(0.99))[[1]] + 0.2 }
  bxp = memDF(variant_id, gene_id) %>%
    ggboxplot(x="ID", y=gene_id, fill="Race", color="Race", add="jitter",
              xlab=variant_id, ylab="Residualized Expression", outlier.
  ↪ shape=NA,
              add.params=list(alpha=0.5), alpha=0.4, legend="bottom",

```

```

        palette="npg", ylim=c(y0,y1),
    ↪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)
}

```

#### 1.1.4 GWAS plotting functions

```

[7]: get_gwas_snps <- function(){
    gwas_snp_file = paste0('/ceph/projects/v4_phase3_paper/inputs/sz_gwas/pgc3/
    ↪',
                                '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(memCAUDATE() %>% inner_join(memGWAS(),
    ↪by=c("variant_id"="our_snp_id")))
}

get_gwas_ordered_snp_df <- function(variant_id, gene_id,
    ↪pgc3_a1_same_as_our_counted, OR){
    df = memDF(variant_id, gene_id)
    if(!pgc3_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,
↪pgc3_a1_same_as_our_counted,
                                OR, title){
  dt = get_gwas_ordered_snp_df(variant_id, gene_id,
↪pgc3_a1_same_as_our_counted, OR)
  bxp = dt %>% mutate_if(is.character, as.factor) %>%
    ggboxplot(x="ID", y=gene_id, fill="Race", color="Race", add="jitter",
              xlab=variant_id, ylab="Residualized Expression", outlier.
↪shape=NA,
              add.params=list(alpha=0.5), alpha=0.4, legend="bottom",
↪#ylim=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

```

[8]: eGenes <- memCAUDATE() %>% arrange(AA, EA) %>% group_by(gene_id) %>% slice(1)
↪%>% arrange(AA, EA)
eGenes %>% head(5)

```

	gene_id <chr>	variant_id <chr>	AA <dbl>	EA <dbl>
	ENSG00000006282.20	chr17:50538753:C:G	0	0
A grouped_df: 5 × 4	ENSG00000010438.16	chr9:33663513:C:T	0	0
	ENSG00000011260.13	chr17:51321627:T:C	0	0
	ENSG00000011376.9	chr3:45514208:A:G	0	0
	ENSG00000013573.16	chr12:31060549:T:C	0	0

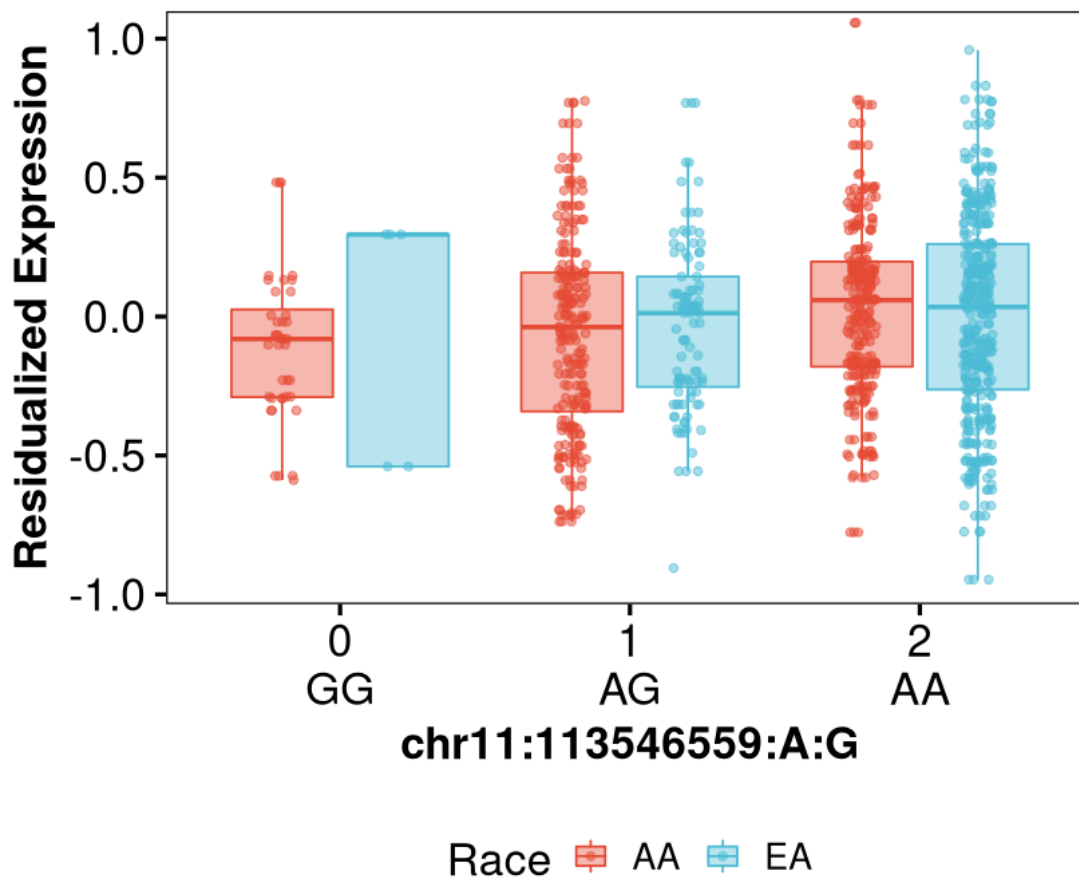
### 1.2.1 DRD2 plot

```

[9]: drd2_id = "ENSG00000149295.13"
drd2_df = memCAUDATE() %>% filter(gene_id == drd2_id)
eqtl_annot = paste("eQTL (AA) lfsr:", signif(drd2_df$AA, 2))
fn = "drd2_eqtl"
plot_simple_eqtl(fn, drd2_df$gene_id, drd2_df$variant_id, eqtl_annot)

```

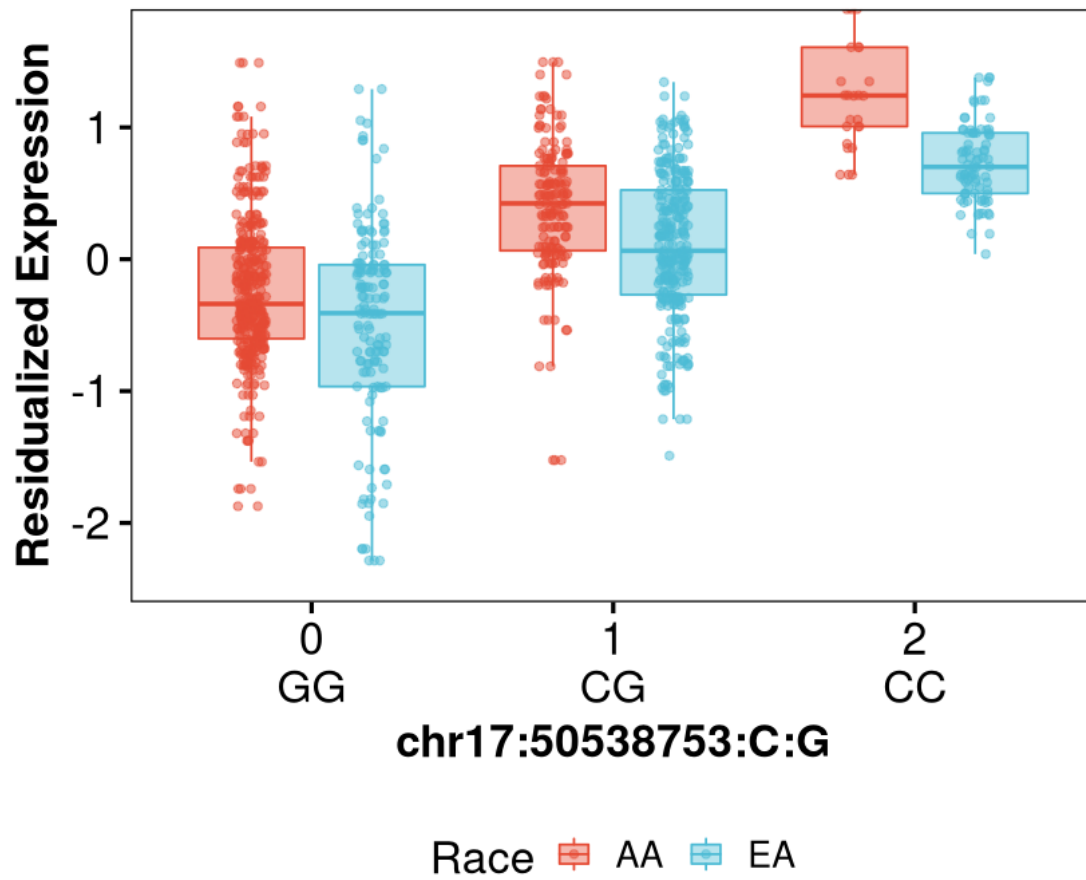
**DRD2**  
**ENSG00000149295.13**  
**eQTL (AA) lfsr: 0.049**



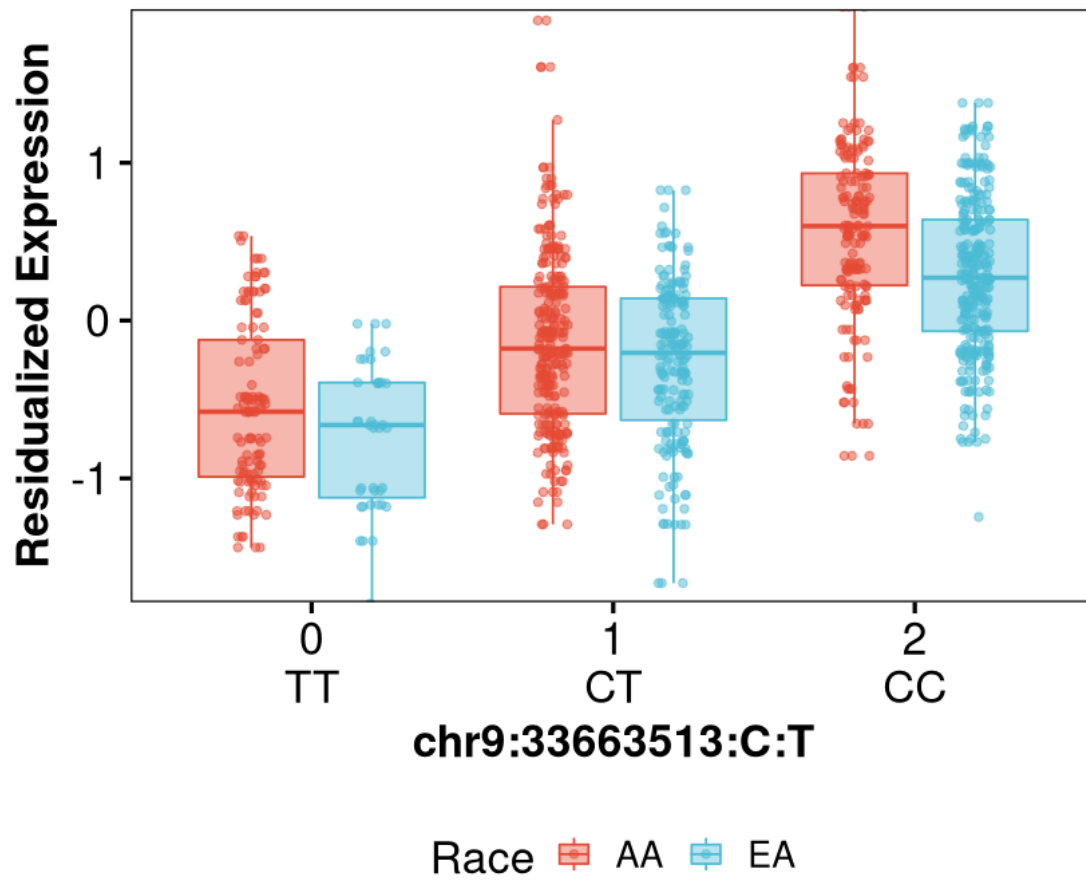
### 1.2.2 Top 5 eQTLs

```
[10]: for(num in 1:5){
      variant_id = eGenes$variant_id[num]
      gene_id = eGenes$gene_id[num]
      eqtl_annot = paste("eQTL (AA & EA) lfsr <", signif(eGenes$AA[num], 2))
      fn = paste0("top_", num, "_eqtl")
      plot_simple_eqtl(fn, gene_id, variant_id, eqtl_annot)
    }
```

**SPATA20**  
**ENSG00000006282.20**  
**eQTL (AA & EA) lfsr < 0**

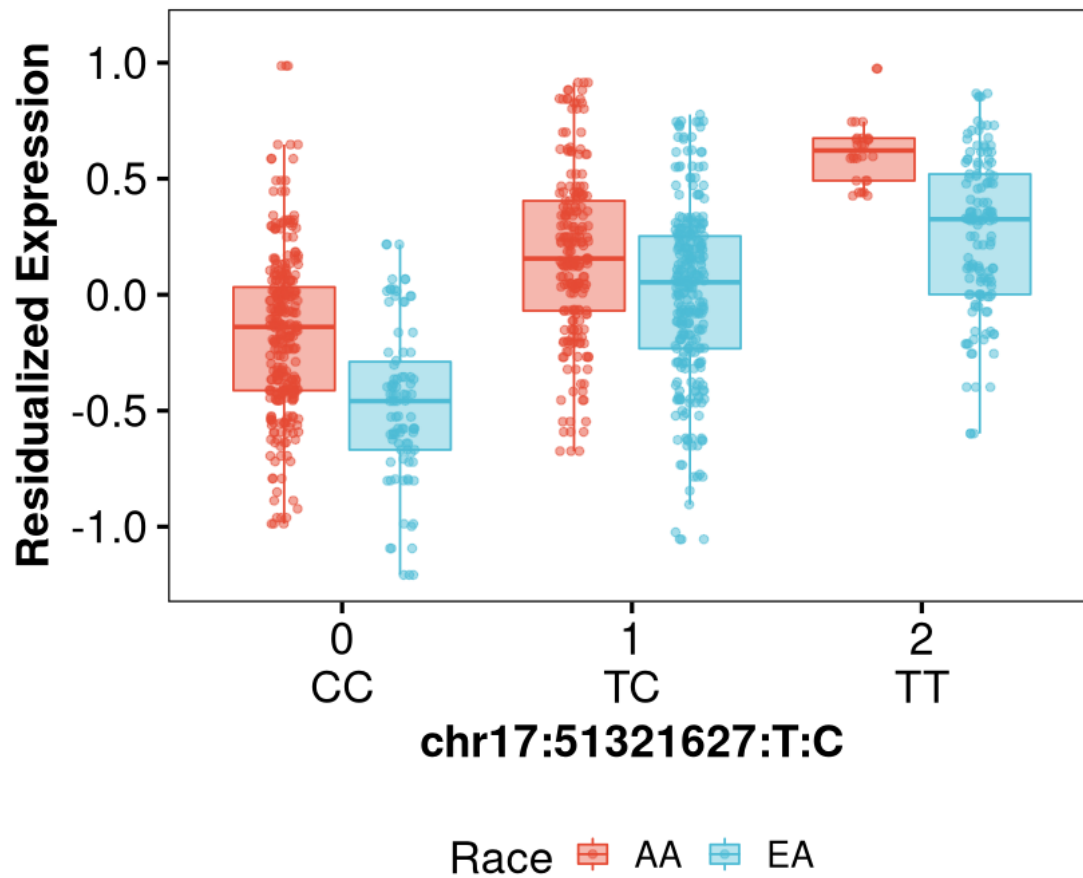


**PRSS3**  
**ENSG00000010438.16**  
**eQTL (AA & EA) lfsr < 0**

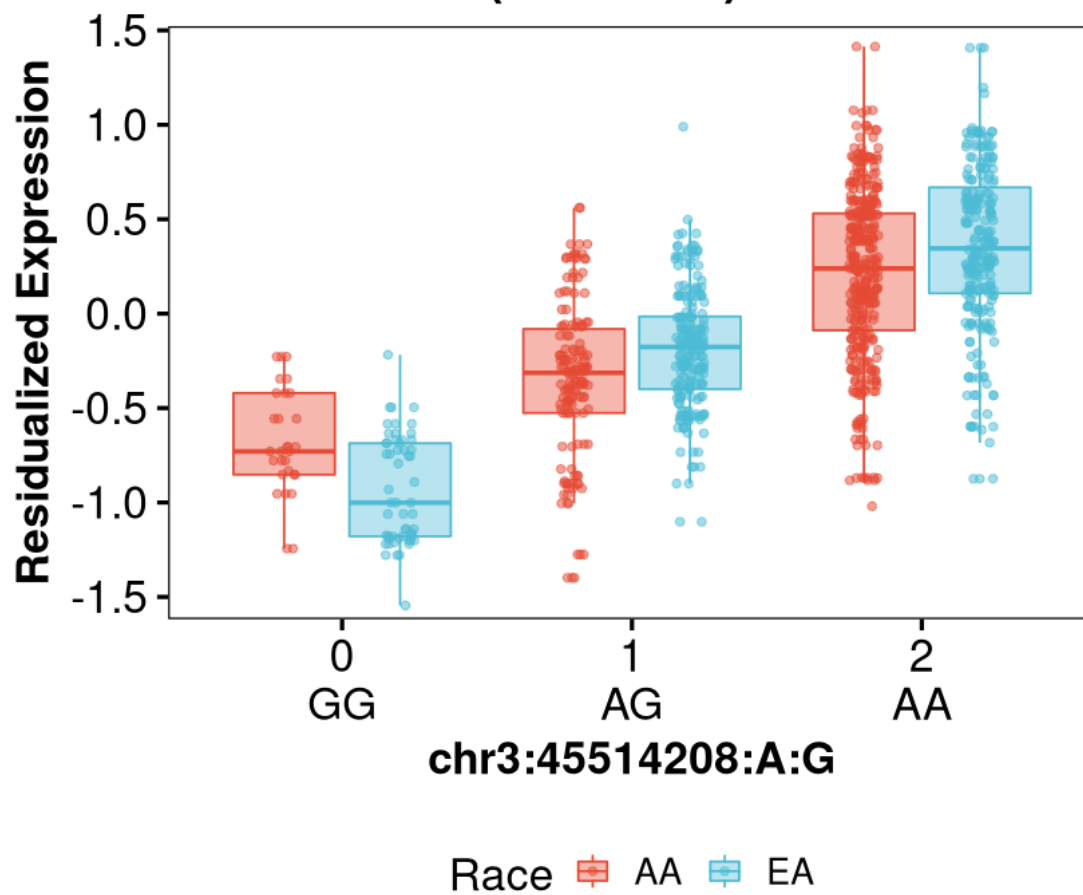


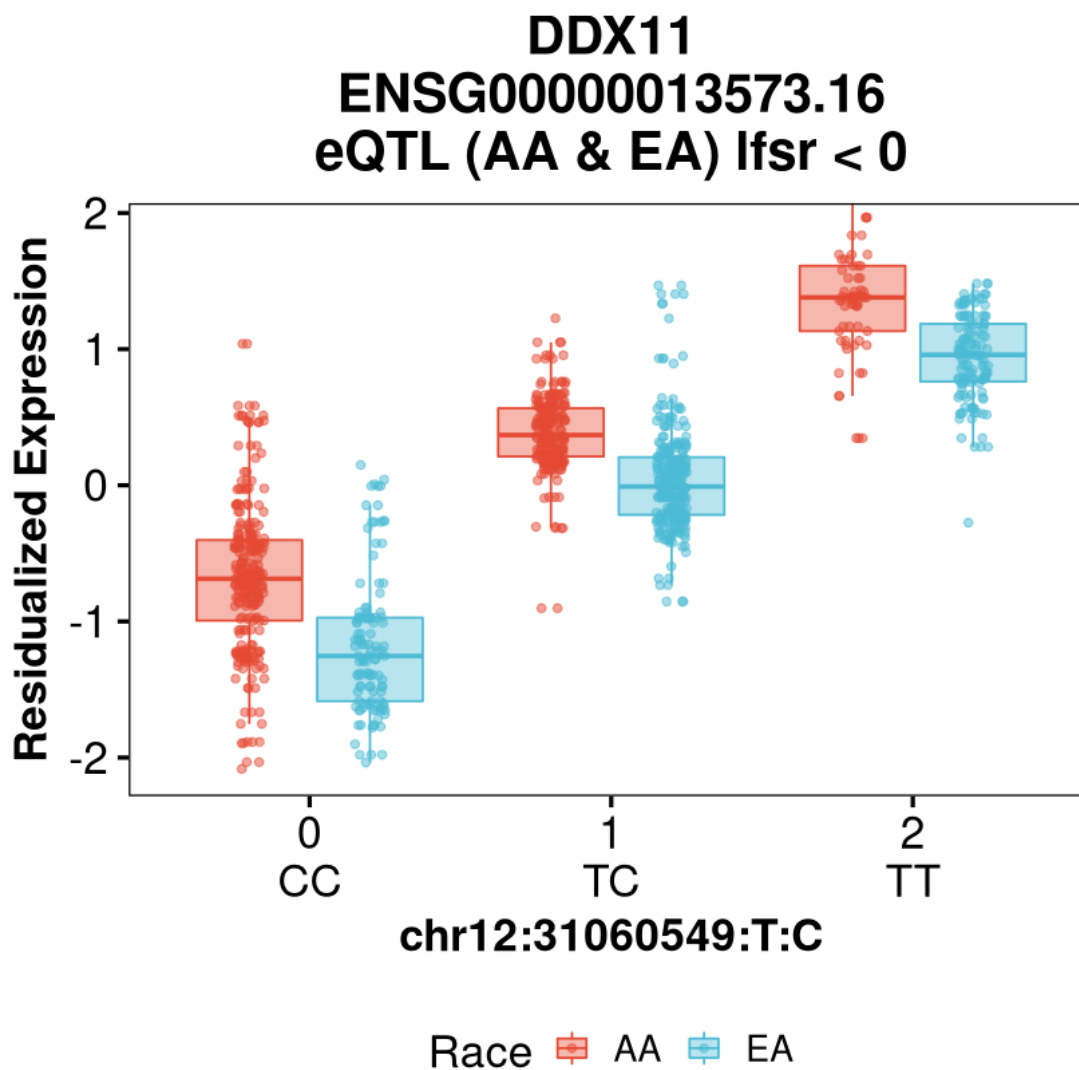


**UTP18**  
**ENSG00000011260.13**  
**eQTL (AA & EA) lfsr < 0**



**LARS2**  
**ENSG00000011376.9**  
**eQTL (AA & EA) lfsr < 0**





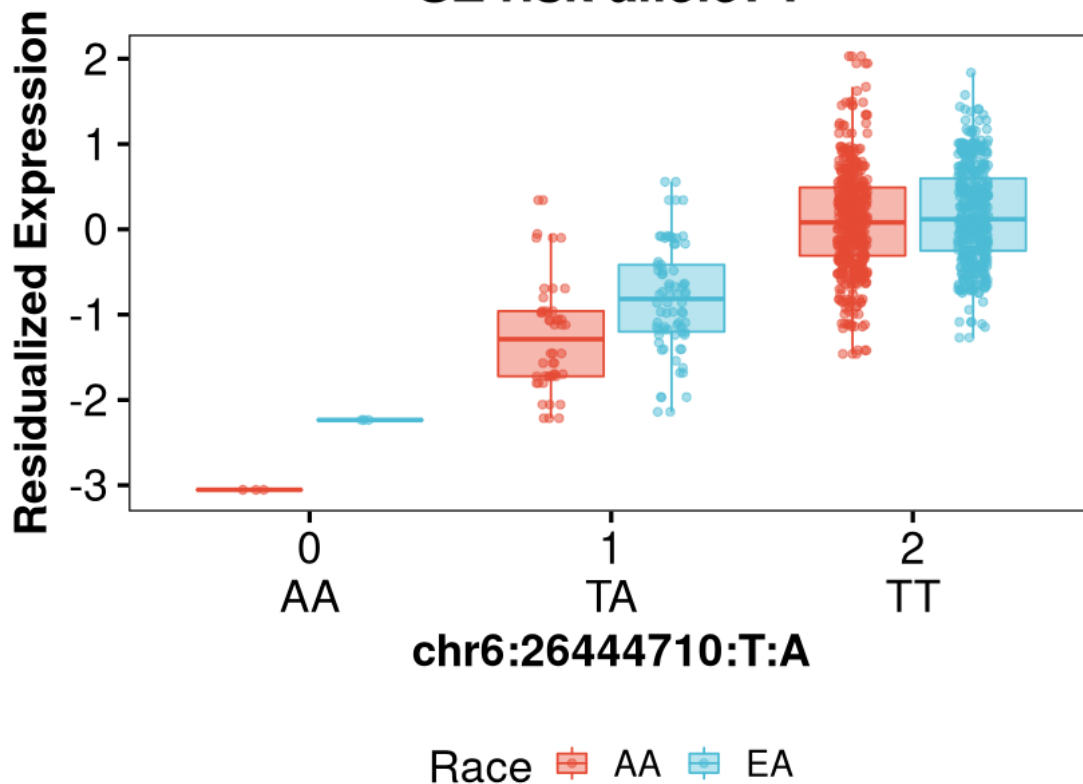
### 1.2.3 Top 5 GWAS associated eQTLs

```
[11]: eGenes_gwas = get_eqtl_gwas_df() %>% arrange(AA, EA, P) %>% group_by(gene_id) %>%
  slice(1) %>% arrange(AA, EA, P)
eGenes_gwas %>% head(5)
```

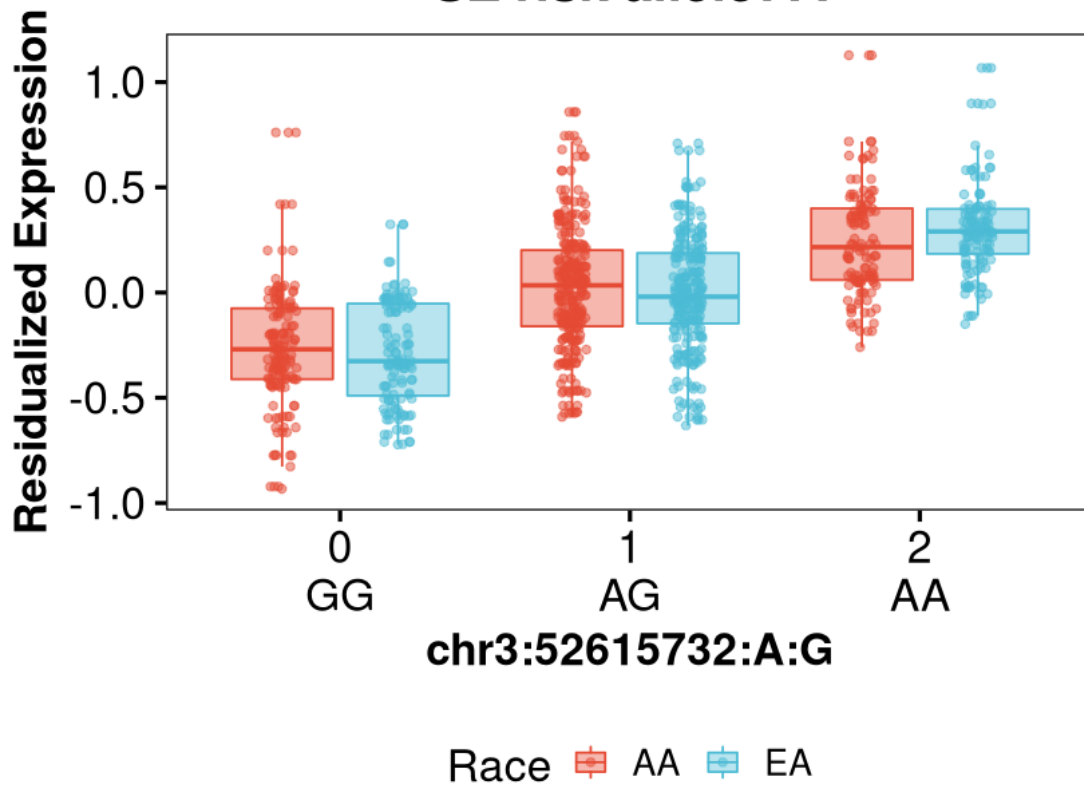
	gene_id <chr>	variant_id <chr>	AA <dbl>	EA <dbl>	V1 <int>	chrN <int>	cm <int>
A grouped_df: 5 × 28	ENSG00000186470.13	chr6:26444710:T:A	0	0	14640	6	0
	ENSG00000163938.16	chr3:52615732:A:G	0	0	10793	3	0
	ENSG00000204428.12	chr6:31742243:C:T	0	0	17671	6	0
	ENSG00000179344.16	chr6:32658238:C:A	0	0	18375	6	0
	ENSG00000196735.11	chr6:32658238:C:A	0	0	18375	6	0

```
[12]: for(num in 1:5){
  fn = paste("top",num,"eqtl_pgc3_variants", sep="_")
  variant_id = eGenes_gwas$variant_id[num]
  gene_id = eGenes_gwas$gene_id[num]
  pgc3_a1_same_as_our_counted = eGenes_gwas$pgc3_a1_same_as_our_counted[num]
  OR = eGenes_gwas$OR[num]
  eqtl_annot = paste("eQTL (AA & EA) lfsr <", signif(eGenes_gwas$AA[num], 2))
  gwas_annot = paste("SZ GWAS pvalue:", signif(eGenes_gwas$P[num], 2))
  risk_annot = paste("SZ risk allele:",
  ↪get_risk_allele(eGenes_gwas$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,
                pgc3_a1_same_as_our_counted, OR, title)
}
```

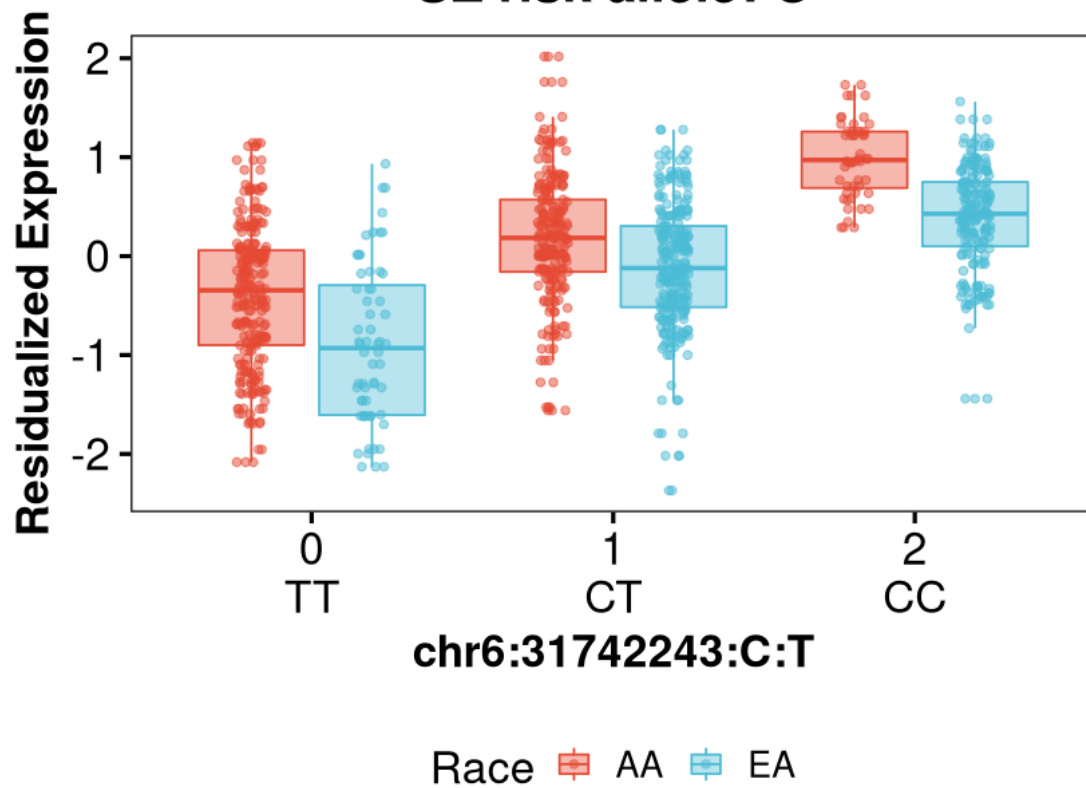
**BTN3A2**  
**ENSG00000186470.13**  
**eQTL (AA & EA) lfsr < 0**  
**SZ GWAS pvalue: 3.4e-23**  
**SZ risk allele: T**



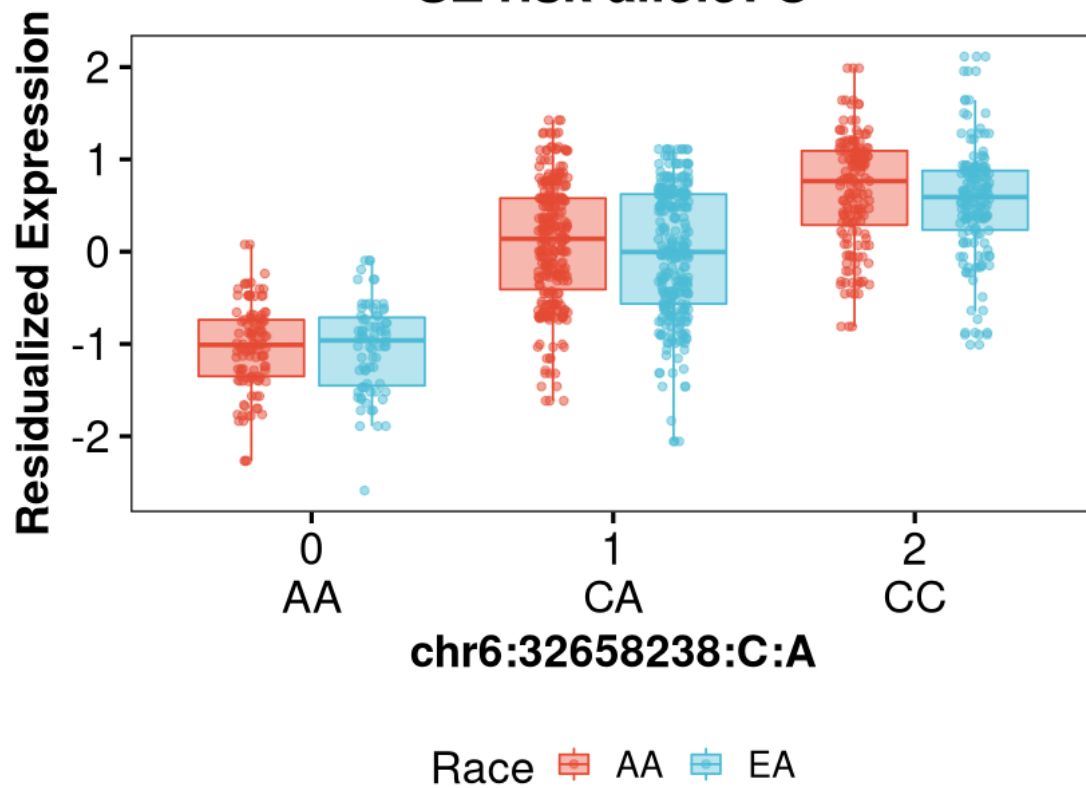
**GNL3**  
**ENSG00000163938.16**  
**eQTL (AA & EA) lfsr < 0**  
**SZ GWAS pvalue: 5.1e-21**  
**SZ risk allele: A**



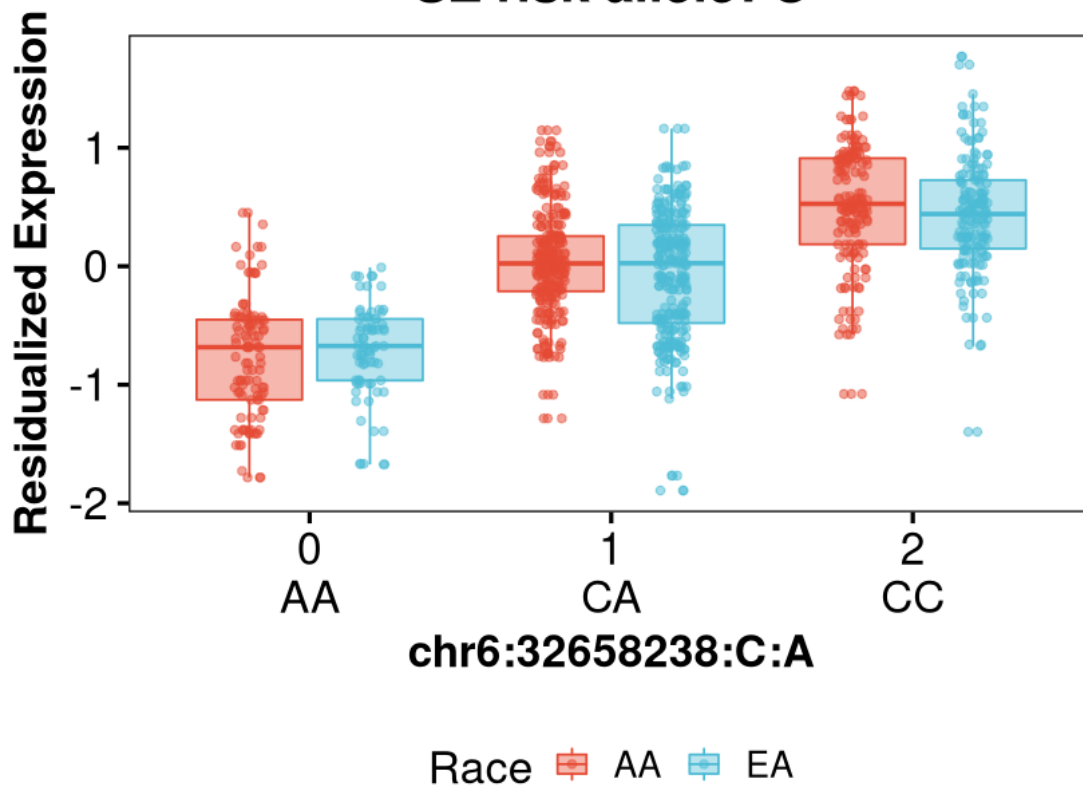
**LY6G5C**  
**ENSG00000204428.12**  
**eQTL (AA & EA) lfsr < 0**  
**SZ GWAS pvalue: 1.9e-14**  
**SZ risk allele: C**



**HLA-DQB1**  
**ENSG00000179344.16**  
**eQTL (AA & EA) lfsr < 0**  
**SZ GWAS pvalue: 1.4e-10**  
**SZ risk allele: C**



**HLA-DQA1**  
**ENSG00000196735.11**  
**eQTL (AA & EA) lfsr < 0**  
**SZ GWAS pvalue: 1.4e-10**  
**SZ risk allele: C**



### 1.3 Session Info

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

```
[1] "2022-03-04 19:50:14 EST"
```

```
      user  system elapsed
8992.041  670.636 1311.906
```

```
$platform $version 'R version 4.1.2 (2021-11-01)'
```

```
$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** '2022-03-04'  
**\$pandoc** '2.14.1 @ /usr/bin/pandoc'

	package <chr>	ondiskversion <chr>	loadedversion <chr>	path <chr>
	abind	1.4.5	1.4.5	/home/jbe
	assertthat	0.2.1	0.2.1	/home/jbe
	backports	1.4.1	1.4.1	/home/jbe
	base64enc	0.1.3	0.1.3	/home/jbe
	broom	0.7.12	0.7.12	/home/jbe
	cachem	1.0.6	1.0.6	/home/jbe
	car	3.0.12	3.0.12	/home/jbe
	carData	3.0.5	3.0.5	/home/jbe
	cellranger	1.1.0	1.1.0	/home/jbe
	cli	3.1.1	3.1.1	/home/jbe
	colorspace	2.0.2	2.0.2	/home/jbe
	crayon	1.4.2	1.4.2	/home/jbe
	data.table	1.14.2	1.14.2	/home/jbe
	DBI	1.1.2	1.1.2	/home/jbe
	dbplyr	2.1.1	2.1.1	/home/jbe
	digest	0.6.29	0.6.29	/home/jbe
	dplyr	1.0.7	1.0.7	/home/jbe
	ellipsis	0.3.2	0.3.2	/home/jbe
	evaluate	0.14	0.14	/home/jbe
	fansi	1.0.2	1.0.2	/home/jbe
	farver	2.1.0	2.1.0	/home/jbe
	fastmap	1.1.0	1.1.0	/home/jbe
	forcats	0.5.1	0.5.1	/home/jbe
	fs	1.5.2	1.5.2	/home/jbe
	generics	0.1.2	0.1.2	/home/jbe
	ggplot2	3.3.5	3.3.5	/home/jbe
	ggpubr	0.4.0	0.4.0	/home/jbe
	ggsci	2.9	2.9	/home/jbe
	ggsignif	0.6.3	0.6.3	/home/jbe
\$packages A packages_info: 78 × 11	glue	1.6.1	1.6.1	/home/jbe
	purrr	0.3.4	0.3.4	/home/jbe
	R.methodsS3	1.8.1	1.8.1	/home/jbe
	R.oo	1.24.0	1.24.0	/home/jbe
	R.utils	2.11.0	2.11.0	/home/jbe
	R6	2.5.1	2.5.1	/home/jbe
	Rcpp	1.0.8	1.0.8	/home/jbe
	readr	2.1.2	2.1.2	/home/jbe
	readxl	1.3.1	1.3.1	/home/jbe
	repr	1.1.4	1.1.4	/home/jbe
	reprex	2.0.1	2.0.1	/home/jbe
	rlang	1.0.0	1.0.0	/home/jbe
	rstatix	0.7.0	0.7.0	/home/jbe
	rstudioapi	0.13	0.13	/home/jbe
	rvest	1.0.2	1.0.2	/home/jbe
	scales	1.1.1	1.1.1	/home/jbe
	sessioninfo	1.2.2	1.2.2	/home/jbe
	stringi	1.7.6	1.7.6	/home/jbe
	stringr	1.4.0	1.4.0	/home/jbe
	svglite	2.0.0	2.0.0	/home/jbe
	systemfonts	1.0.3	1.0.3	/home/jbe