main

September 1, 2021

1 Plotting eQTLs, increase font sizes

1.0.1 Kynon Jade Benjamin and Apuã Paquola

[1]:
```
import re
import functools
import subprocess
import numpy as np
import pandas as pd
from plotnine import *
from pandas_plink import read_plink
from warnings import filterwarnings
from matplotlib.cbook import mplDeprecation

filterwarnings("ignore",category=mplDeprecation)
filterwarnings('ignore', category=UserWarning, module='plotnine.*')
filterwarnings('ignore', category=DeprecationWarning, module='plotnine.*')
```

1.1 Configuration

[2]:
```
tissue = "dlpfc"; feature = "genes"
config = {
    'biomart_file': '../_h/biomart.csv',
    'residual_expression_file': "../../../../prep_eqtl_analysis/%s/%s/
 ↪covariates/" % (tissue, feature)+\
    "residualized_expression/_m/%s_residualized_expression.csv" % feature,
    'phenotype_file': '/ceph/projects/v4_phase3_paper/inputs/phenotypes/_m/
 ↪merged_phenotypes.csv',
    'plink_file_prefix': '/ceph/projects/v4_phase3_paper/inputs/genotypes/_m/
 ↪LIBD_Brain_TopMed',
    'eqtl_output_file': '../../../summary_table/_m/
 ↪Brainseq_sex_interacting_4features_3regions.eFeatures.txt.gz',
    'gwas_snp_file': '/ceph/projects/v4_phase3_paper/inputs/sz_gwas/pgc2_clozuk/
 ↪map_phase3/_m/libd_hg38_pgc2sz_snps_p5e_minus8.tsv'
}
```

1.2 Functions

1.2.1 Expression functions

[3]:
```
@functools.lru_cache()
def tissue_map(tissue):
    return {"caudate": "Caudate", "dlpfc": "DLPFC",
            "hippocampus": "Hippocampus"}[tissue]


@functools.lru_cache()
def feature_map(feature):
    return {"genes": "Gene", "transcripts": "Transcript",
            "exons": "Exon", "junctions": "Junction"}[feature]


@functools.lru_cache()
def get_biomart_df():
    biomart = pd.read_csv(config['biomart_file'], index_col=0)
    biomart['description'] = biomart['description'].str.replace('\[Source.
↪*$','', regex=True)
    return biomart


@functools.lru_cache()
def get_residual_expression_df():
    return pd.read_csv(config['residual_expression_file'], index_col=0).
↪transpose()


@functools.lru_cache()
def get_pheno_df():
    return pd.read_csv(config['phenotype_file']).set_index("BrNum").loc[:,␣
↪["RNum", "Sex", "Dx"]]


@functools.lru_cache()
def get_expression_and_pheno_df():
    return pd.merge(get_pheno_df(), get_residual_expression_df(),
                    left_index=True, right_index=True)


@functools.lru_cache()
def get_gene_id_df():
    return pd.DataFrame({'gene_id': get_residual_expression_df().columns,
                         'ensembl_gene_id': get_residual_expression_df().
↪columns.str.replace('\..+$','', regex=True)})
```

```
@functools.lru_cache()
def gene_info_from_symbol(gene_symbol):
    return␣
↪get_biomart_df()[get_biomart_df()['external_gene_name']==gene_symbol]\
        .merge(get_gene_id_df(), on='ensembl_gene_id', how='left')


@functools.lru_cache()
def gene_id_from_symbol(gene_symbol):
    df = gene_info_from_symbol(gene_symbol)
    assert df.shape[0] == 1
    return df[['gene_id']].iloc[0].values[0]
```

1.2.2 Genotype and eQTL functions

[4]:
```
def letter_snp(number, a0, a1):
    '''
    Example:
    letter_snp(0, 'A', 'G') is 'AA'
    letter_snp(1, 'A', 'G') is 'AG'
    letter_snp(2, 'A', 'G') is 'GG'
    '''
    if np.isnan(number):
        return np.nan
    if len(a0)==1 and len(a1)==1:
        sep = ''
    else:
        sep = ' '
    return sep.join(sorted([a0]*int(number) + [a1]*(2-int(number))))


@functools.lru_cache()
def get_plink_tuple():
    '''
    Usage: (bim, fam, bed) = get_plink_tuple()
    '''
    return read_plink(config['plink_file_prefix'])


@functools.lru_cache()
def get_eFeature_df():
    eqtl_df = pd.read_csv(config["eqtl_output_file"], sep='\t')
    return eqtl_df[(eqtl_df["Type"] == feature_map(feature)) &
                   (eqtl_df["Tissue"] == tissue_map(tissue))]
```

```
@functools.lru_cache()
def get_gwas_snps():
    return pd.read_csv(config['gwas_snp_file'], sep='\t', index_col=0)


@functools.lru_cache()
def get_risk_allele(snp_id):
    gwas_snp = get_gwas_snp(snp_id)
    if gwas_snp['OR'].iloc[0] > 1:
        ra = gwas_snp['A1'].iloc[0]
    else:
        ra = gwas_snp['A2'].iloc[0]
    return ra


@functools.lru_cache()
def get_snp_df(snp_id):
    '''
    Returns a dataframe containing the genotype on snp snp_id.
    The allele count is the same as in the plink files.

    Example:
    get_snp_df('rs653953').head(5)

            rs653953_num rs653953_letter rs653953
    Br5168             0              GG    0\nGG
    Br2582             1              AG    1\nAG
    Br2378             1              AG    1\nAG
    Br5155             2              AA    2\nAA
    Br5182             2              AA    2\nAA
    '''
    (bim, fam, bed) = get_plink_tuple()
    brain_ids = list(set(get_expression_and_pheno_df().index).
 ↪intersection(set(fam['fid'])))
    snp_info = bim[bim['snp']==snp_id]
    snp_pos = snp_info.iloc[0]['i']
    fam_pos = list(fam.drop_duplicates(subset="fid").set_index('fid').
 ↪loc[brain_ids]['i'])
    dfsnp = (pd.DataFrame(bed[[snp_pos]].compute()[:,fam_pos],
                          columns=brain_ids, index=[snp_id + '_num'])
             .transpose().dropna())
    my_letter_snp = functools.partial(letter_snp, a0=snp_info.iloc[0]['a0'],␣
 ↪a1=snp_info.iloc[0]['a1'])
    dfsnp[[snp_id + '_num']] = 2 - dfsnp[[snp_id + '_num']].astype('int')
    dfsnp[snp_id + '_letter'] = dfsnp[snp_id + '_num'].apply(my_letter_snp)
    dfsnp[snp_id] = (dfsnp[snp_id + '_num'].astype('str') + '\n' +
                     dfsnp[snp_id + '_letter'].astype('str')).astype('category')
    return dfsnp


@functools.lru_cache()
def get_gwas_ordered_snp_df(snp_id):
    '''
    Returns a dataframe containing the genotype on snp snp_id.
    The allele count is the number of risk alleles according to GWAS.

    Example:
    get_gwas_ordered_snp_df('rs653953').head(5)

            rs653953_num rs653953_letter rs653953
    Br5168             2              GG    2\nGG
    Br2582             1              AG    1\nAG
    Br2378             1              AG    1\nAG
    Br5155             0              AA    0\nAA
    Br5182             0              AA    0\nAA
    '''
    pgc = get_gwas_snps()
    dfsnp = get_snp_df(snp_id).copy()
    gwas_snp = get_gwas_snp(snp_id)
    if gwas_snp['pgc2_a1_same_as_our_counted'].iloc[0]:
        if gwas_snp['OR'].iloc[0] > 1:
            pass
        else:
            dfsnp[[snp_id + '_num']] = 2 - dfsnp[[snp_id + '_num']]
    else:
        if gwas_snp['OR'].iloc[0] > 1:
            dfsnp[[snp_id + '_num']] = 2 - dfsnp[[snp_id + '_num']]
        else:
            pass
    dfsnp[snp_id] = (dfsnp[snp_id + '_num'].astype('str') + '\n' +
                     dfsnp[snp_id + '_letter'].astype('str')).astype('category')
    return dfsnp
```

1.2.3 Plotting functions

```
[5]: def get_snp_gene_pheno_df(snp_id, gene_id, snp_df_func):
         pheno_columns = list(get_pheno_df().columns)
         expr_df = get_expression_and_pheno_df()[pheno_columns + [gene_id]]
         snp_df =  snp_df_func(snp_id)
         return expr_df.merge(snp_df, left_index=True, right_index=True)



     def simple_snp_expression_plot_impl(snp_id, gene_id, snp_df_func):
         df = get_snp_gene_pheno_df(snp_id, gene_id, snp_df_func)
```

```
    y0 = df[gene_id].quantile(.01) - 0.26
    y1 = df[gene_id].quantile(.99) + 0.26
    p = ggplot(df, aes(x=snp_id, y=gene_id, fill='Sex')) \
    + geom_boxplot(alpha=0.4, outlier_alpha=0) \
    + geom_jitter(position=position_jitterdodge(jitter_width=0.25),
                  stroke=0, alpha=0.6) \
    + ylim(y0, y1) \
    + theme_bw(base_size=15) \
    + theme(panel_grid=element_blank(),
            axis_title=element_text(face="bold"))
    return p


def simple_snp_expression_plot(snp_id, gene_id):
    return simple_snp_expression_plot_impl(snp_id, gene_id, get_snp_df)


def simple_gwas_ordered_snp_expression_plot(snp_id, gene_id):
    return simple_snp_expression_plot_impl(snp_id, gene_id,␣
 ↪get_gwas_ordered_snp_df)


def get_gene_symbol(gene_id, biomart=get_biomart_df()):
    ensge = re.sub('\..+$','', gene_id)
    ggg = biomart[biomart['ensembl_gene_id']==ensge]
    if ggg.shape[0]==0:
        return '', ''
    gs = ggg['external_gene_name'].values[0]
    de = ggg['description'].values[0]
    if type(de)!=str:
        de = ''
    de = re.sub('\[Source:.*$','',de)
    return gs, de


def get_gwas_snp(snp_id):
    gwas = get_gwas_snps()
    r = gwas[gwas['our_snp_id']==snp_id]
    assert len(r) == 1
    return r


def gwas_annotation(snp_id):
    return 'SZ GWAS pvalue: %.1e' % get_gwas_snp(snp_id).iloc[0]['P']


def eqtl_annotation(snp_id, gene_id):
    eqtl_df = get_eFeature_df()
    r = eqtl_df[(eqtl_df['variant_id']==snp_id) & (eqtl_df['gene_id']==gene_id)]
    assert len(r)==1
    return 'eQTL adjusted p-value: %.1e' % r.iloc[0]['BF']


def risk_allele_annotation(snp_id):
    return 'SZ risk allele: %s' % get_risk_allele(snp_id)


def annotated_eqtl_plot(snp_id, gene_id):
    p = simple_snp_expression_plot(snp_id, gene_id)
    gene_symbol, gene_description = get_gene_symbol(gene_id)
    title ="\n".join([gene_symbol,
                      eqtl_annotation(snp_id, gene_id)
                      ])
    p += ggtitle(title) + ylab('Residualized Expression')
    return p


def gwas_annotated_eqtl_plot(snp_id, gene_id):
    p = simple_gwas_ordered_snp_expression_plot(snp_id, gene_id)
    gene_symbol, gene_description = get_gene_symbol(gene_id)
    title ="\n".join([gene_symbol,
                      eqtl_annotation(snp_id, gene_id),
                      gwas_annotation(snp_id),
                      risk_allele_annotation(snp_id)
                      ])
    p += ggtitle(title) + ylab('Residualized Expression')
    return p


def save_plot(p, fn):
    for ext in ['png', 'pdf', 'svg']:
        p.save(fn + '.' + ext)
```

1.3 Plot eQTLs

1.3.1 DRD2

[6]:
```
get_eFeature_df()[(get_eFeature_df()["gene_id"] == gene_id_from_symbol('DRD2'))]
```

[6]:
```
Empty DataFrame
Columns: [variant_id, gene_id, gencodeID, slope, statistic, pval_nominal, BF,
eigenMT_BH, TESTS, Type, Tissue]
Index: []
```

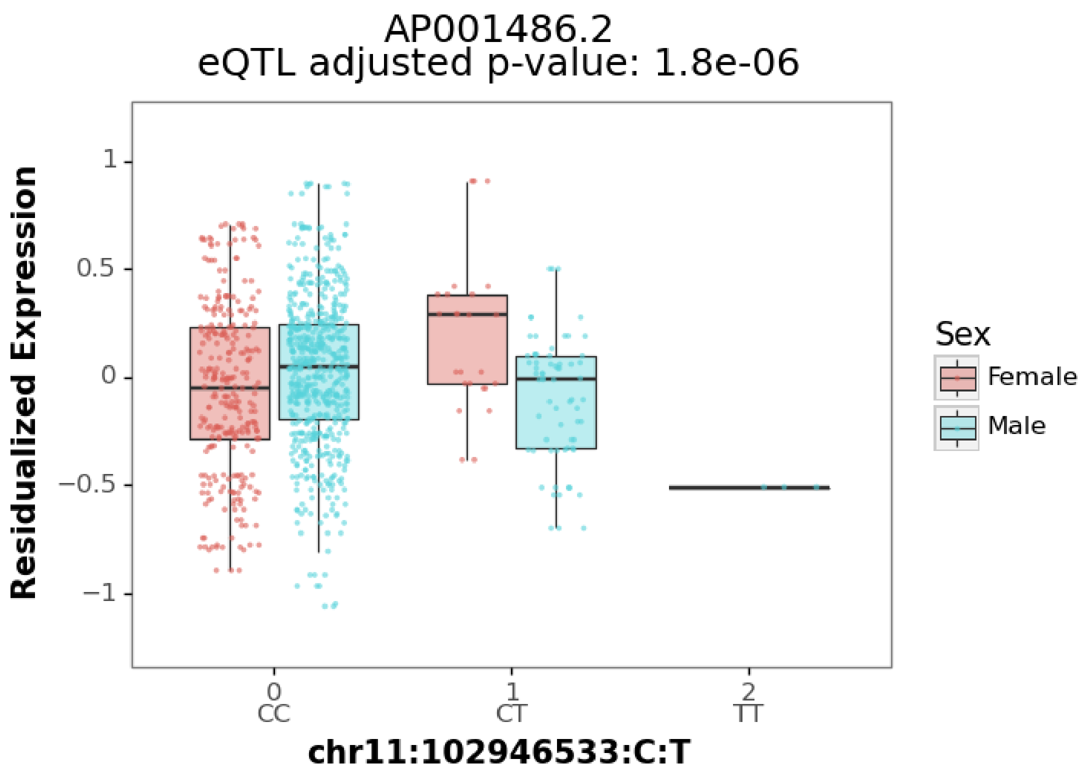
1.3.2 Top 5 eQTLs

[7]:
```
eqtl_df = get_eFeature_df()
eqtl_df.head()
```

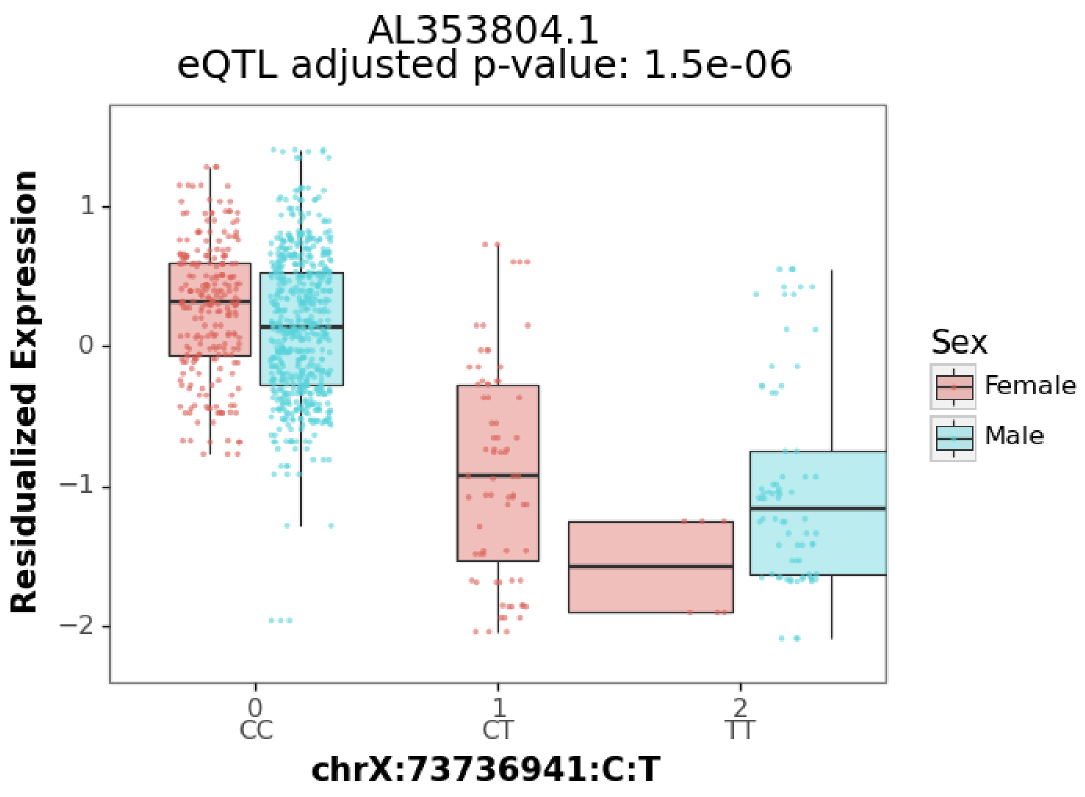
[7]:
```
                    variant_id             gene_id           gencodeID  \
29020      chr1:196175713:C:T  ENSG00000000971.15  ENSG00000000971.15
29021       chrX:65756506:T:A  ENSG00000001497.16  ENSG00000001497.16
29022      chr4:11426453:A:AG   ENSG00000002587.9   ENSG00000002587.9
29023      chr2:200747038:C:T  ENSG00000003400.14  ENSG00000003400.14
29024  chr1:22575250:CTTATTA:C  ENSG00000004487.15  ENSG00000004487.15

          slope  statistic  pval_nominal        BF  eigenMT_BH  TESTS  Type  \
29020  0.380554  14.718097      0.000132  0.020039    0.586787    152  Gene
29021 -0.454968  -5.143391      0.000160  0.011876    0.560779     74  Gene
29022 -0.732540  -8.886183      0.000042  0.027774    0.612625    667  Gene
29023  0.455473  10.846644      0.000148  0.044193    0.656748    298  Gene
29024  0.432105  11.118354      0.000077  0.036023    0.646335    466  Gene

      Tissue
29020  DLPFC
29021  DLPFC
29022  DLPFC
29023  DLPFC
29024  DLPFC
```

[8]:
```
top_5 = eqtl_df.sort_values('pval_nominal').reset_index(drop=True).head(5)
for x in top_5.itertuples():
    filename = "top_%d_eqtl_%s" % (x.Index, tissue)
    p = annotated_eqtl_plot(x.variant_id, x.gene_id)
    print(filename, x.Index, x.variant_id, x.gene_id)
    print(p)
    save_plot(p, filename)
```

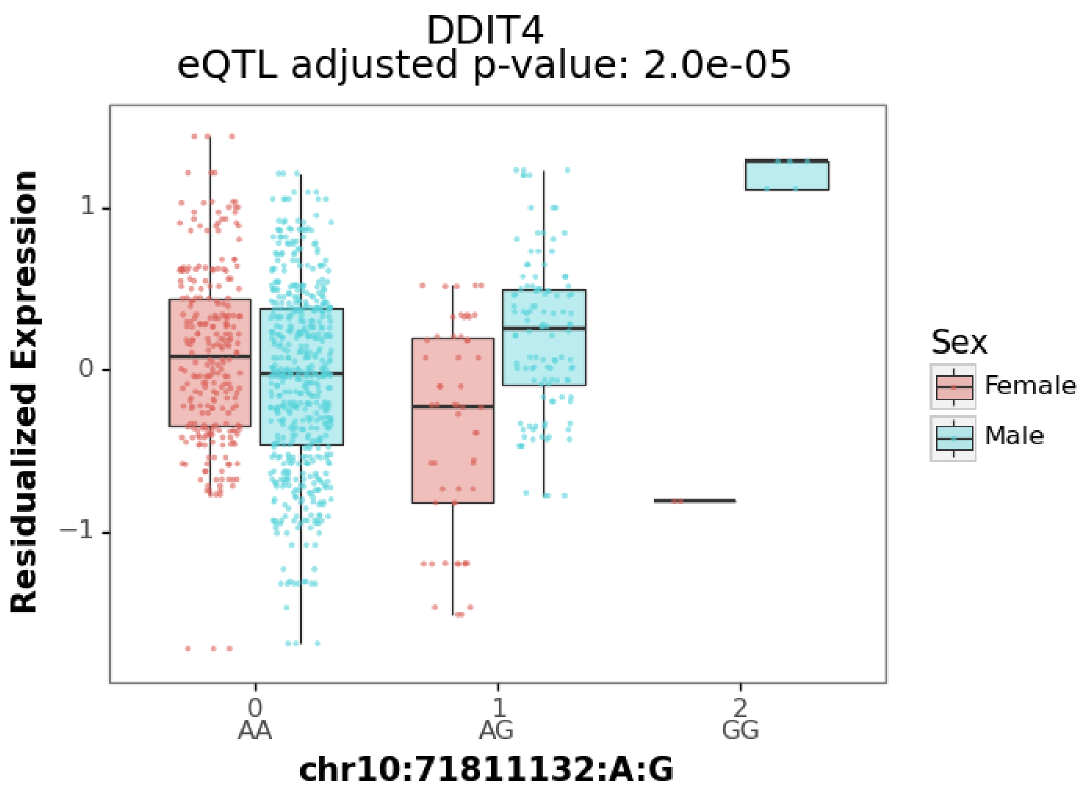
```
Mapping files: 100%|      | 3/3 [00:26<00:00,  8.76s/it]

top_0_eqtl_dlpfc 0 chr11:102946533:C:T ENSG00000260966.1
```



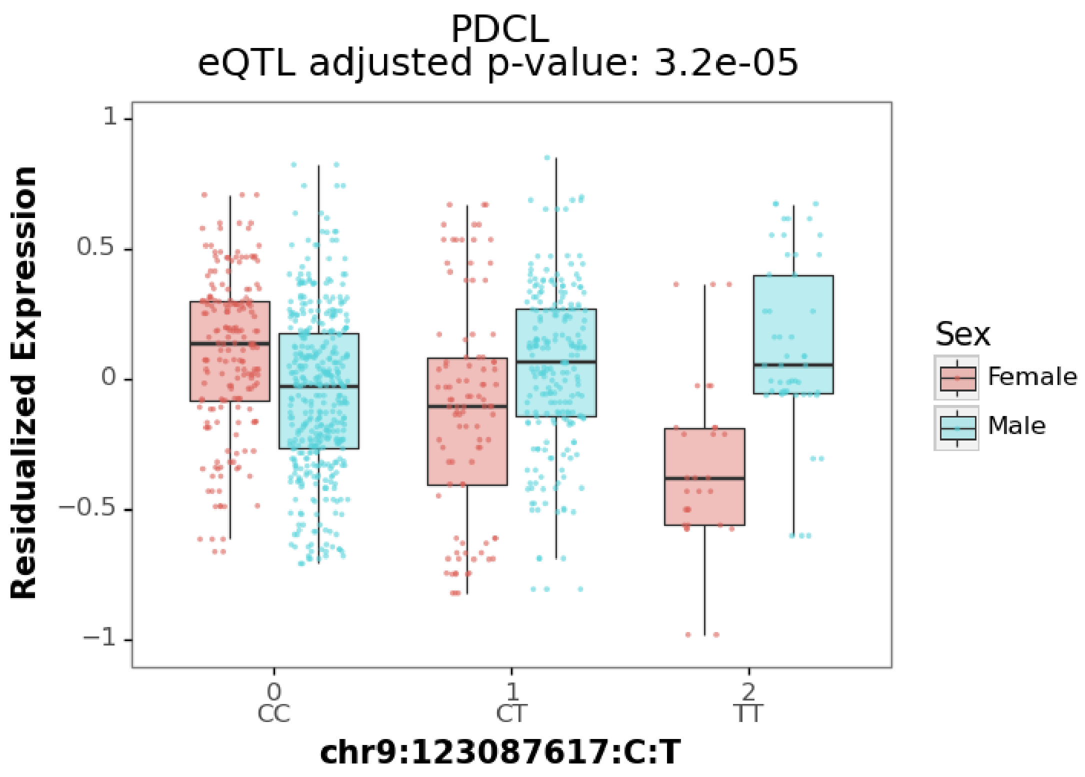
```
<ggplot: (8739662760394)>
top_1_eqtl_dlpfc 1 chrX:73736941:C:T ENSG00000228906.1
```



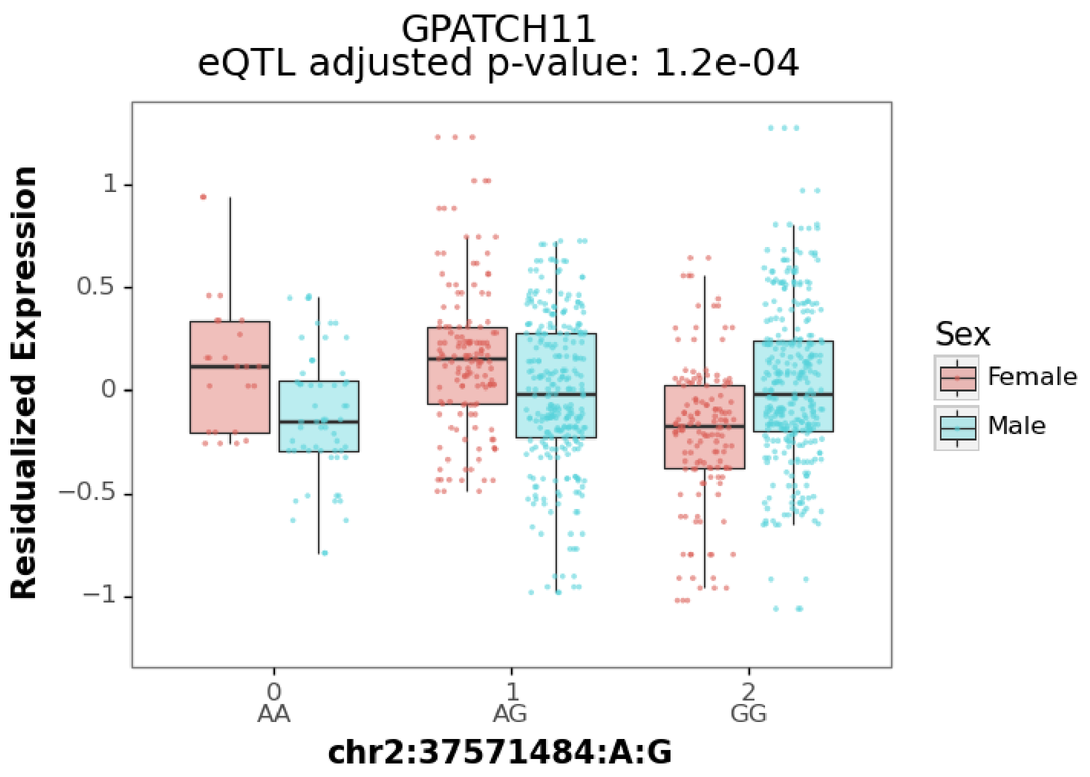
```
<ggplot: (8739662559507)>
top_2_eqtl_dlpfc 2 chr10:71811132:A:G ENSG00000168209.4
```



```
<ggplot: (8739636181008)>
top_3_eqtl_dlpfc 3 chr9:123087617:C:T ENSG00000136940.13
```



```
<ggplot: (8739636066545)>
top_4_eqtl_dlpfc 4 chr2:37571484:A:G ENSG00000152133.14
```

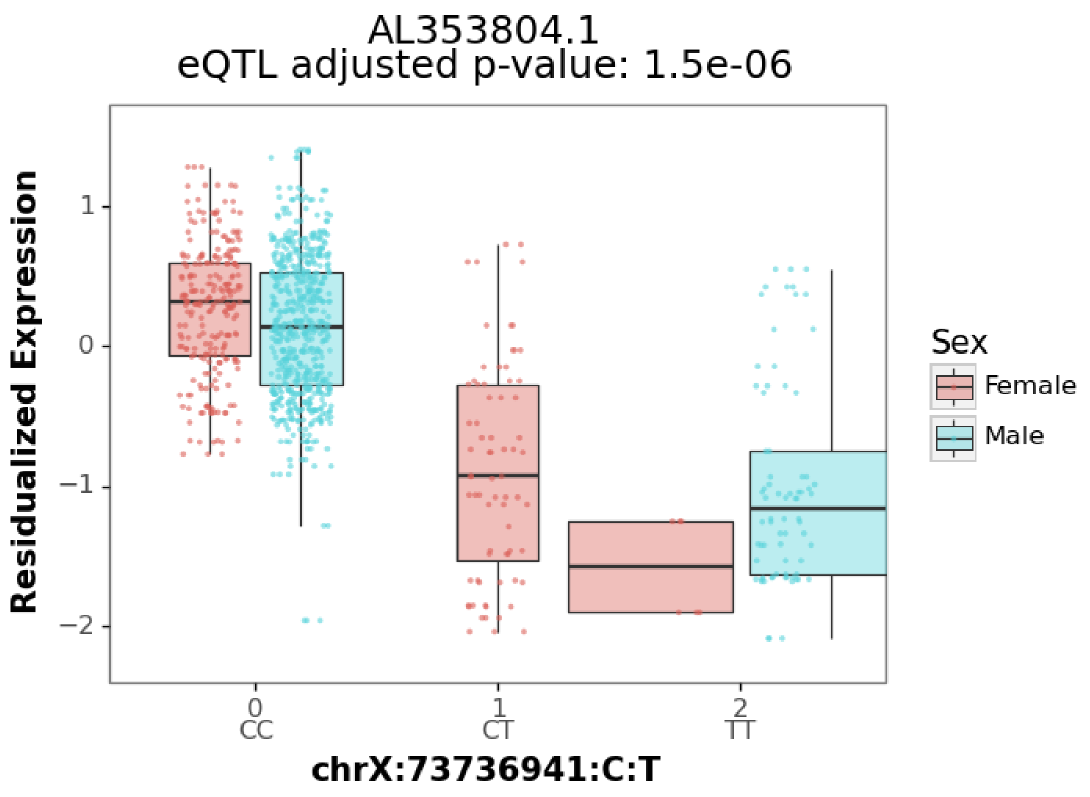


```
<ggplot: (8739635715780)>
```

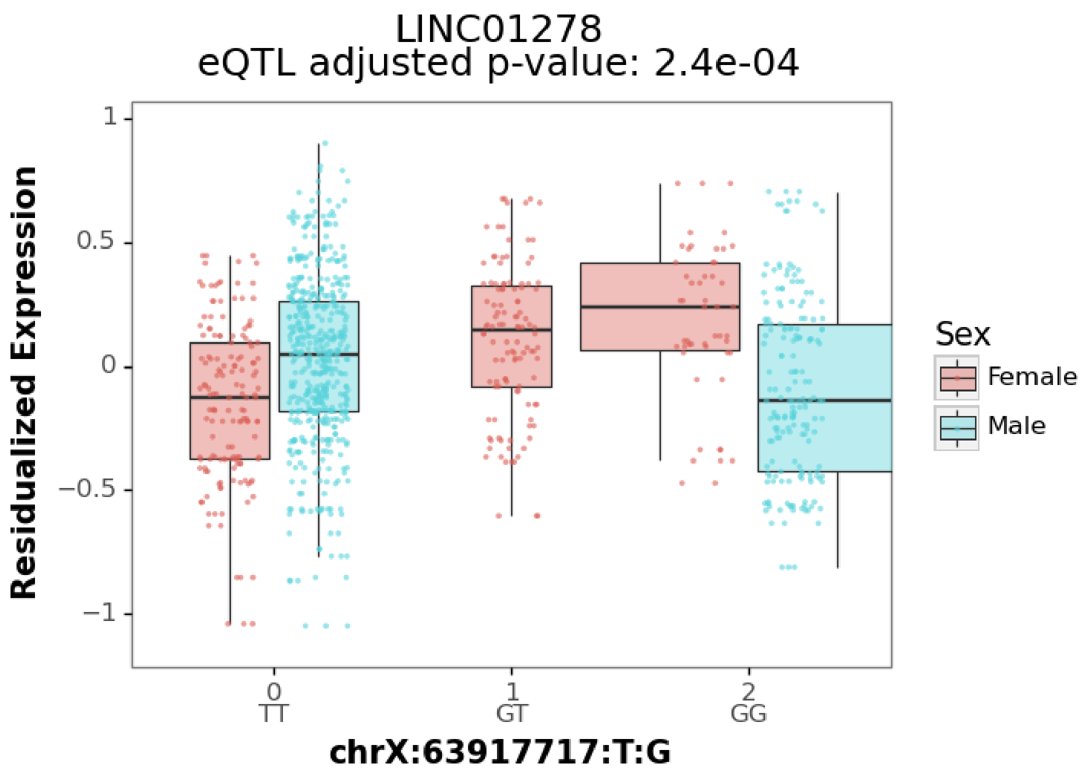
1.3.3 Top 5 X-linked genes

```
[9]: top_5_x = eqtl_df[eqtl_df['variant_id'].str.contains("chrX")].
      ↪sort_values("pval_nominal").reset_index(drop=True).head(5)
     for x  in top_5_x.itertuples():
         filename = "top_%d_eqtl_xlinked_%s" % (x.Index, tissue)
         p = annotated_eqtl_plot(x.variant_id, x.gene_id)
         print(filename, x.Index, x.variant_id, x.gene_id)
         print(p)
         save_plot(p, filename)
```

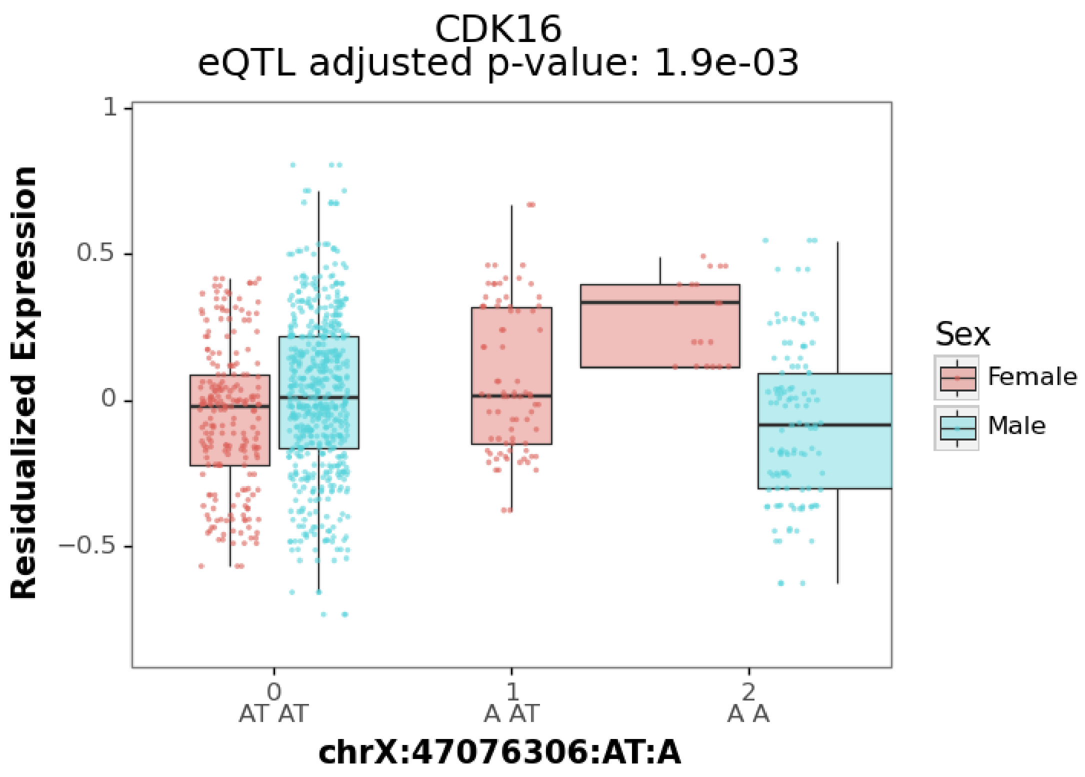
```
top_0_eqtl_xlinked_dlpfc 0 chrX:73736941:C:T ENSG00000228906.1
```



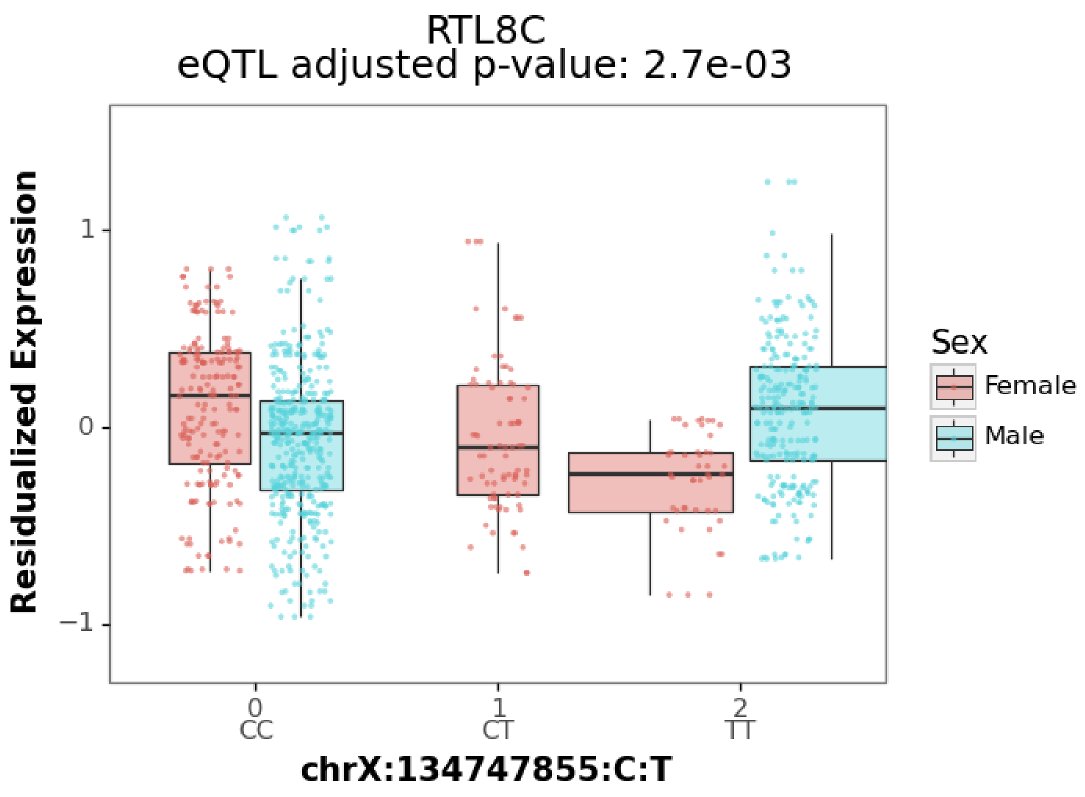
```
<ggplot: (8739764125609)>
top_1_eqtl_xlinked_dlpfc 1 chrX:63917717:T:G ENSG00000235437.7
```



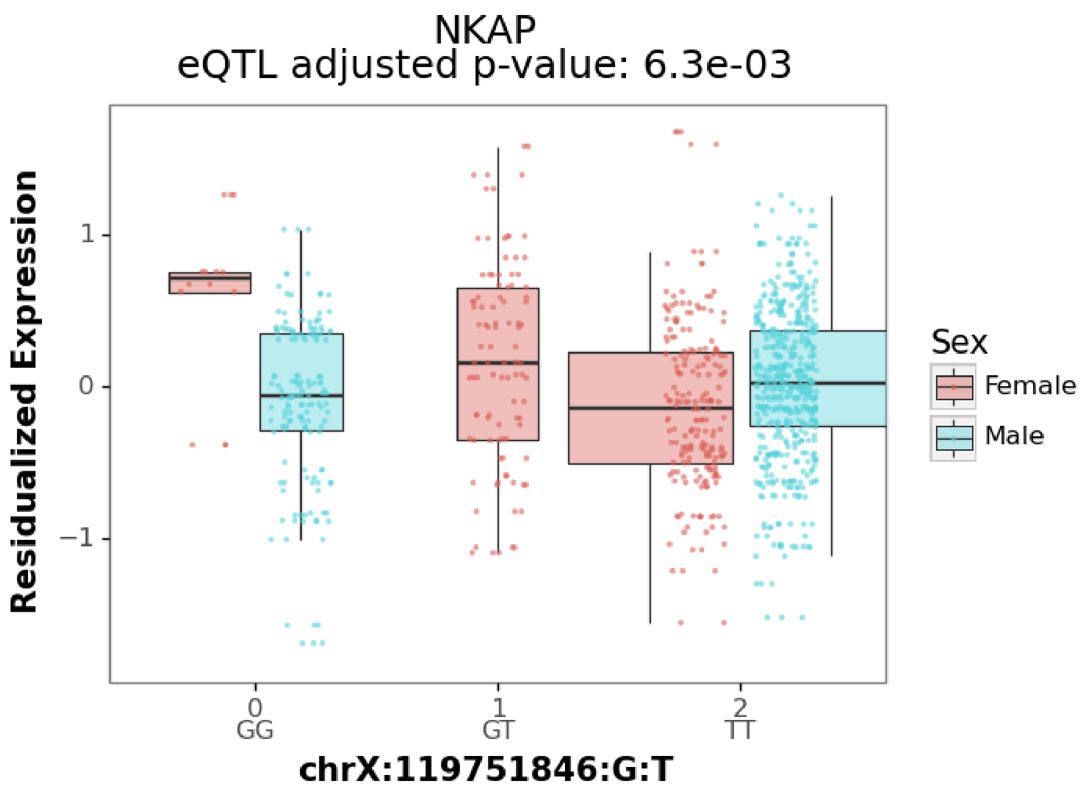
```
<ggplot: (8739662709242)>
top_2_eqtl_xlinked_dlpfc 2 chrX:47076306:AT:A ENSG00000102225.15
```



```
<ggplot: (8739662717992)>
top_3_eqtl_xlinked_dlpfc 3 chrX:134747855:C:T ENSG00000134590.13
```



```
<ggplot: (8739662229450)>
top_4_eqtl_xlinked_dlpfc 4 chrX:119751846:G:T ENSG00000101882.9
```



```
<ggplot: (8739663360139)>
```

1.3.4 Top 5 eQTL with GWAS significant index SNP

```
[10]: gwas_eqtl_df = eqtl_df.merge(get_gwas_snps(), left_on = 'variant_id',
                                 right_on = 'our_snp_id', suffixes=['','_gwas'])
      print(gwas_eqtl_df.shape)
      gwas_eqtl_df.head()
```

```
(6, 33)
```

```
[10]:          variant_id             gene_id           gencodeID     slope  \
      0  chr10:102875591:C:T  ENSG00000156374.14  ENSG00000156374.14  0.408080
      1    chr6:31888293:G:A  ENSG00000204371.11  ENSG00000204371.11  0.360059
      2    chr6:32225442:C:T   ENSG00000243649.8   ENSG00000243649.8 -0.664658
      3    chr6:31494358:G:A   ENSG00000244731.7   ENSG00000244731.7 -0.762762
      4  chr8:110618046:TG:T   ENSG00000254241.1   ENSG00000254241.1 -0.274868

         statistic  pval_nominal        BF  eigenMT_BH  TESTS  Type  …  A2  \
      0  10.794880      0.000149  0.047714    0.666100    320  Gene  …   T
      1  11.418709      0.000004  0.001779    0.434507    502  Gene  …   A
      2 -14.774041      0.000015  0.008005    0.535673    525  Gene  …   T
```

```
3 -11.406318      0.000018  0.009065    0.539284    498  Gene  …   A
4 -10.339230      0.000113  0.019356    0.585158    171  Gene  …  TG

        OR        SE             P  hg19chrc hg38chrc    hg38pos  \
0  1.08930  0.011207  2.270000e-14     chr10    chr10  102875591
1  0.92611  0.010290  8.620000e-14      chr6     chr6   31888293
2  1.11790  0.013264  4.480000e-17      chr6     chr6   32225442
3  1.18310  0.015322  5.000000e-28      chr6     chr6   31494358
4  1.08730  0.012600  3.060000e-11      chr8     chr8  110618046

  pgc2_a1_same_as_our_counted        rsid  is_index_snp
0                       False  rs12765002         False
1                       False    rs486416         False
2                       False   rs3130295         False
3                       False   rs3130923         False
4                        True  rs35771435         False

[5 rows x 33 columns]
```

[11]:
```
top_gwas_eqtl_df = gwas_eqtl_df[(gwas_eqtl_df['is_index_snp'])].
 ↪sort_values(['BF', 'P'])
print(top_gwas_eqtl_df.shape)
top_gwas_eqtl_df.head()
```

```
(0, 33)
```

[11]:
```
Empty DataFrame
Columns: [variant_id, gene_id, gencodeID, slope, statistic, pval_nominal, BF,
eigenMT_BH, TESTS, Type, Tissue, chrN, our_snp_id, cm, pos, our_counted,
our_alt, chrom, SNP, Freq.A1, CHR, BP, A1, A2, OR, SE, P, hg19chrc, hg38chrc,
hg38pos, pgc2_a1_same_as_our_counted, rsid, is_index_snp]
Index: []

[0 rows x 33 columns]
```

[12]:
```
top_gwas_eqtl_df = gwas_eqtl_df.sort_values(['BF', 'P']).reset_index(drop=True)
print(top_gwas_eqtl_df.shape)
top_gwas_eqtl_df.head(10)
```

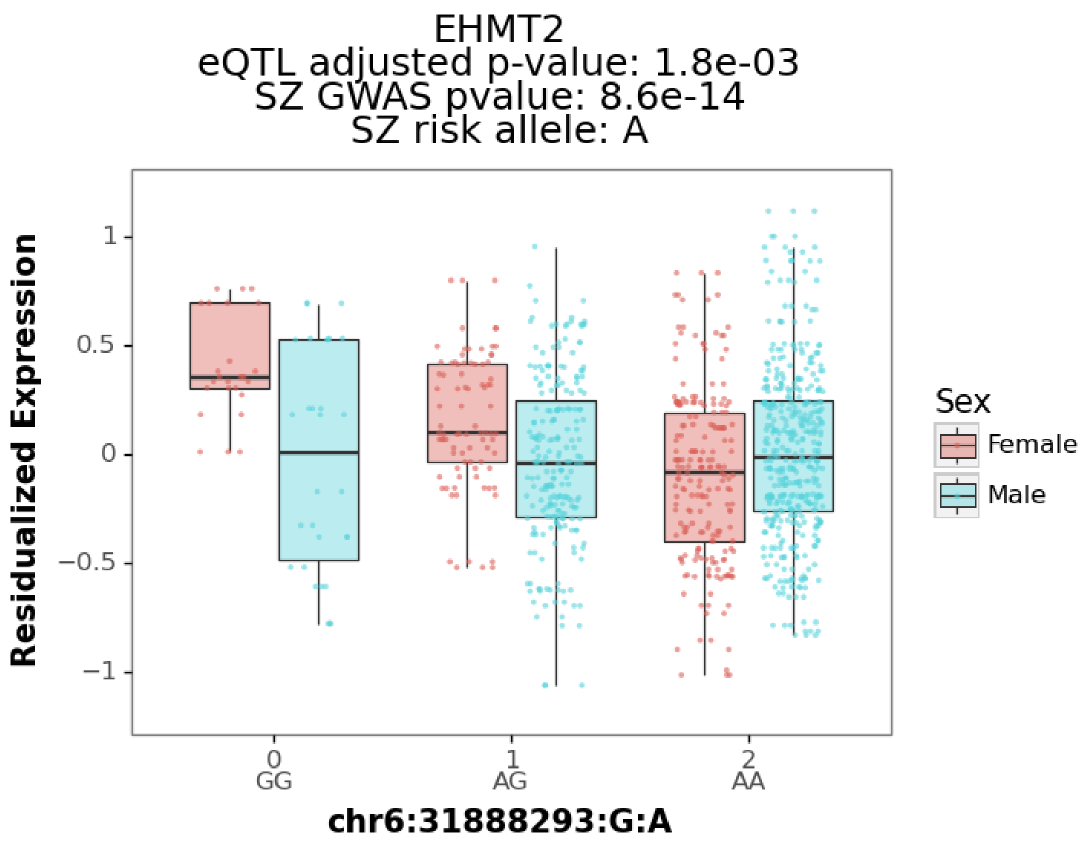
```
(6, 33)
```

[12]:
```
            variant_id             gene_id           gencodeID     slope  \
0    chr6:31888293:G:A  ENSG00000204371.11  ENSG00000204371.11  0.360059
1    chr6:32225442:C:T   ENSG00000243649.8   ENSG00000243649.8 -0.664658
2    chr6:31494358:G:A   ENSG00000244731.7   ENSG00000244731.7 -0.762762
3    chr6:26573403:C:G   ENSG00000274290.2   ENSG00000274290.2 -0.553277
4  chr8:110618046:TG:T   ENSG00000254241.1   ENSG00000254241.1 -0.274868
5  chr10:102875591:C:T  ENSG00000156374.14  ENSG00000156374.14  0.408080
```

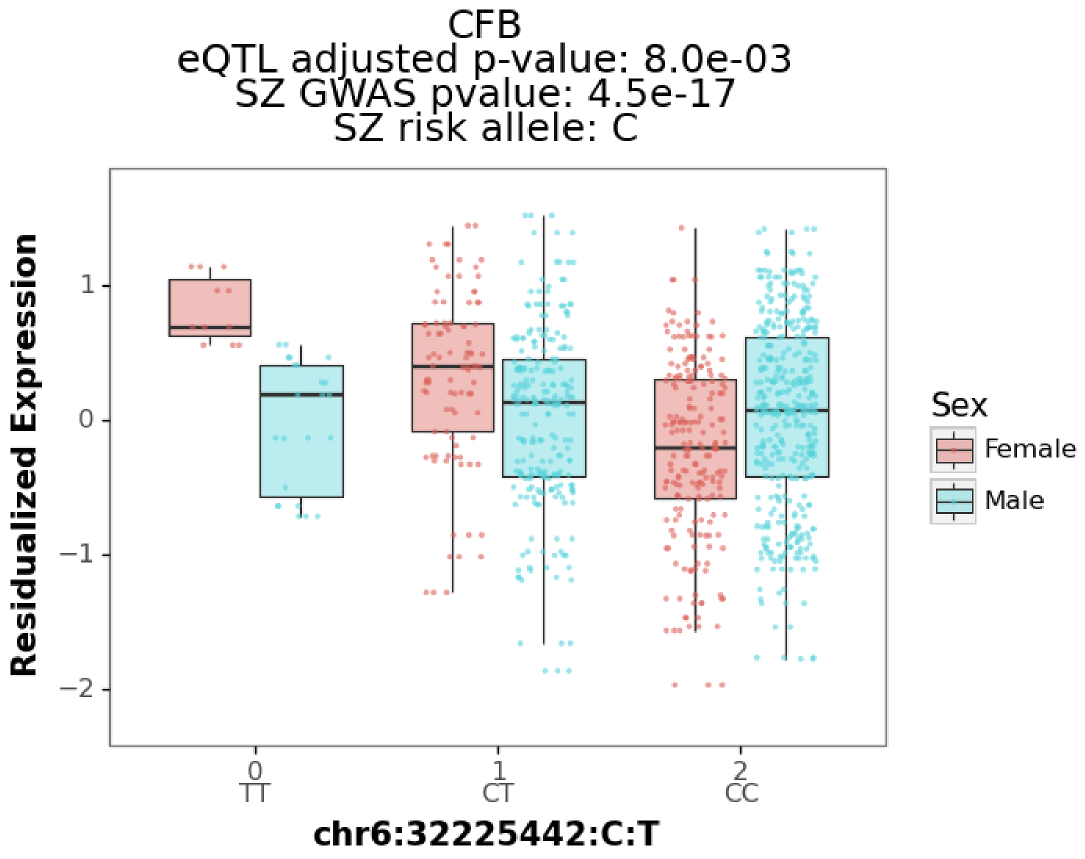
```
   statistic  pval_nominal        BF  eigenMT_BH  TESTS  Type  …  A2  \
0  11.418709      0.000004  0.001779    0.434507    502  Gene  …   A
1 -14.774041      0.000015  0.008005    0.535673    525  Gene  …   T
2 -11.406318      0.000018  0.009065    0.539284    498  Gene  …   A
3 -10.810298      0.000046  0.012635    0.560779    276  Gene  …   G
4 -10.339230      0.000113  0.019356    0.585158    171  Gene  …  TG
5  10.794880      0.000149  0.047714    0.666100    320  Gene  …   T

        OR        SE             P  hg19chrc hg38chrc    hg38pos  \
0  0.92611  0.010290  8.620000e-14      chr6     chr6   31888293
1  1.11790  0.013264  4.480000e-17      chr6     chr6   32225442
2  1.18310  0.015322  5.000000e-28      chr6     chr6   31494358
3  1.16530  0.014746  3.300000e-25      chr6     chr6   26573403
4  1.08730  0.012600  3.060000e-11      chr8     chr8  110618046
5  1.08930  0.011207  2.270000e-14     chr10    chr10  102875591

  pgc2_a1_same_as_our_counted        rsid  is_index_snp
0                       False    rs486416         False
1                       False   rs3130295         False
2                       False   rs3130923         False
3                       False  rs13214027         False
4                        True  rs35771435         False
5                       False  rs12765002         False

[6 rows x 33 columns]
```

[13]:
```
top_5_gwas = top_gwas_eqtl_df.head(5)
for x  in top_5_gwas.itertuples():
    filename = "top_%d_eqtl_in_gwas_significant_snps_%s" % (x.Index, tissue)
    p = gwas_annotated_eqtl_plot(x.variant_id, x.gene_id)
    print(filename, x.Index, x.variant_id, x.gene_id)
    print(p)
    save_plot(p, filename)
```

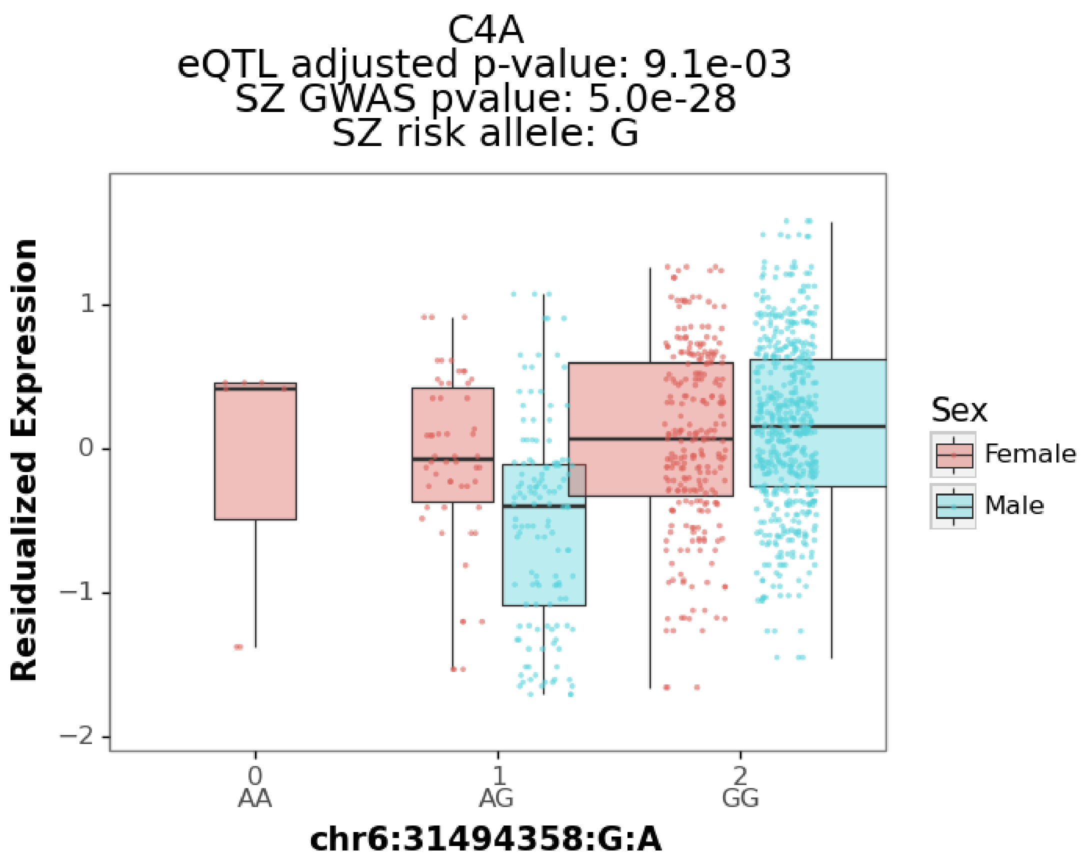
```
top_0_eqtl_in_gwas_significant_snps_dlpfc 0 chr6:31888293:G:A ENSG00000204371.11
```



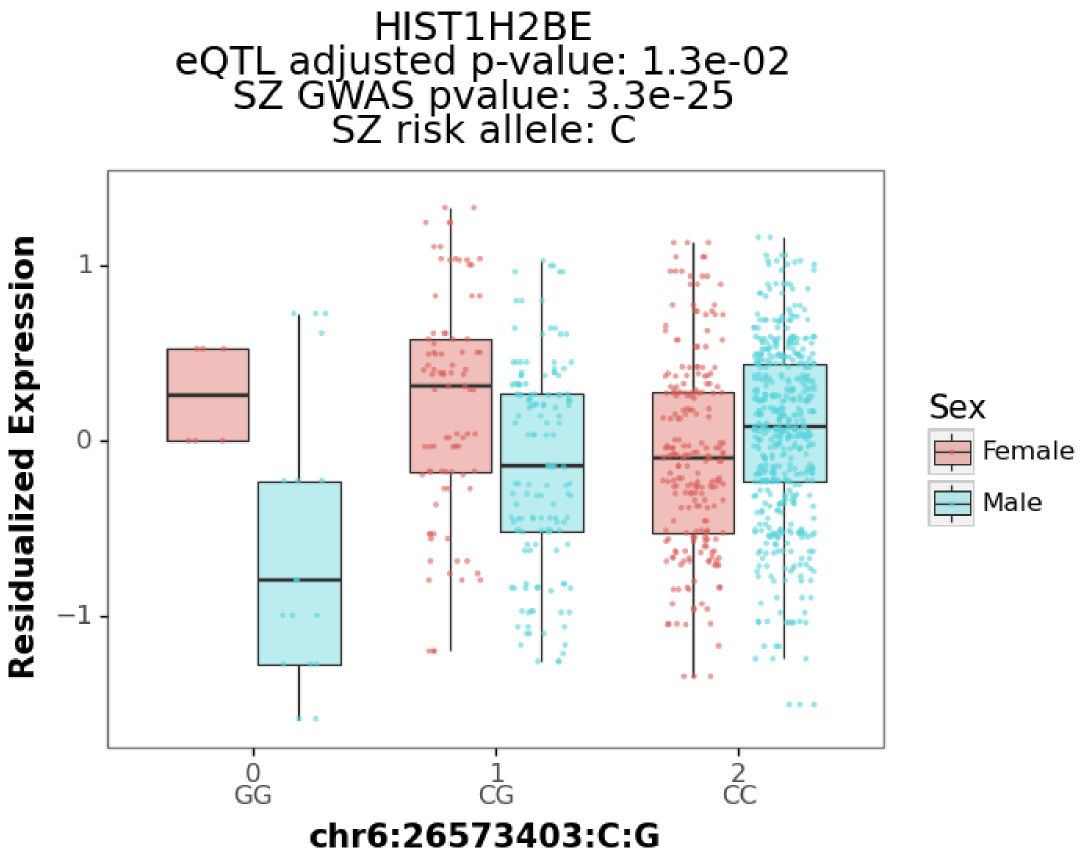
```
<ggplot: (8739663415432)>
top_1_eqtl_in_gwas_significant_snps_dlpfc 1 chr6:32225442:C:T ENSG00000243649.8
```



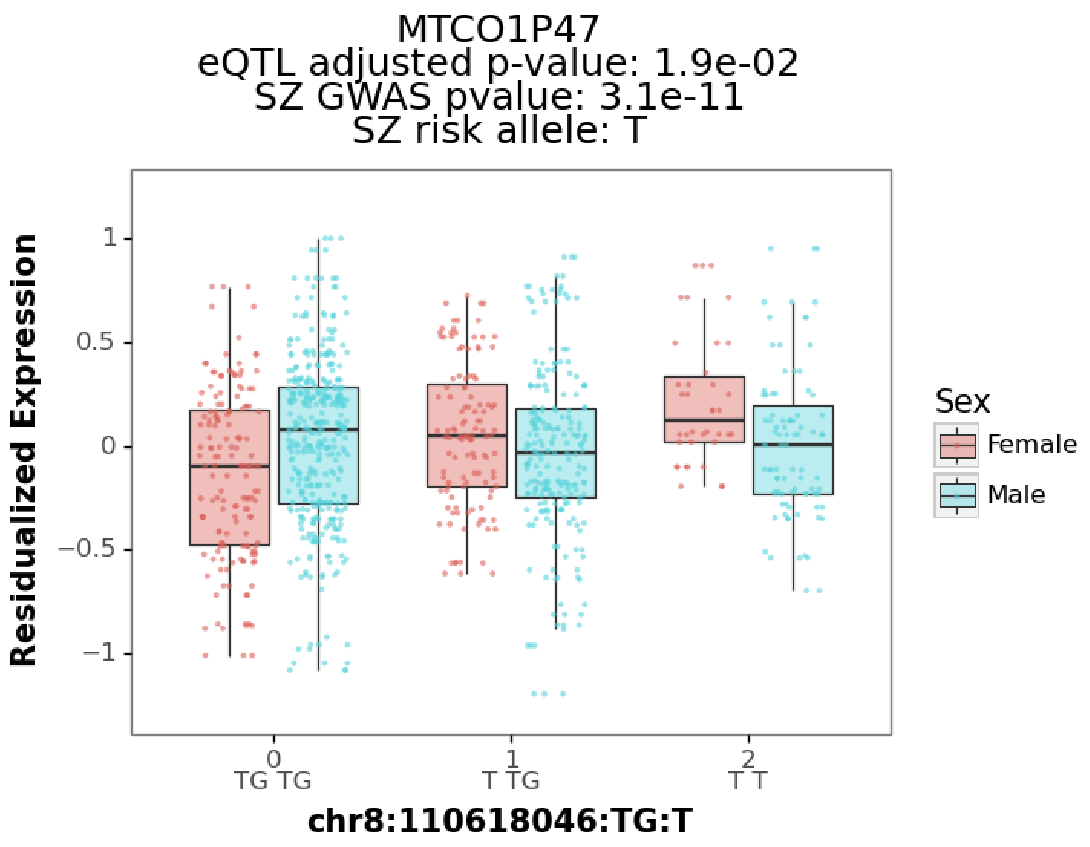
```
<ggplot: (8739634910085)>
top_2_eqtl_in_gwas_significant_snps_dlpfc 2 chr6:31494358:G:A ENSG00000244731.7
```



```
<ggplot: (8739635303355)>
top_3_eqtl_in_gwas_significant_snps_dlpfc 3 chr6:26573403:C:G ENSG00000274290.2
```



```
<ggplot: (8739636188145)>
top_4_eqtl_in_gwas_significant_snps_dlpfc 4 chr8:110618046:TG:T
ENSG00000254241.1
```


MTCO1P47
eQTL adjusted p-value: 1.9e-02
SZ GWAS pvalue: 3.1e-11
SZ risk allele: T
Residualized Expression
1
0.5
0
−0.5
−1
0
TG TG
1
T TG
2
T T
chr8:110618046:TG:T
Sex
Female
Male

```
<ggplot: (8739635459151)>
```

[ ]: