main_genes

September 2, 2021

1 Enrichment and Overlap of PGC2+CLOZUK

[1]:
```
import re
import os, errno
import functools
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
from scipy.stats import fisher_exact, binom_test

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

1.1 Config and Functions

[2]:
```
config = {
    'biomart_file': '../_h/biomart.csv',
    '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',
    'gwas_snp_file': '/ceph/projects/v4_phase3_paper/inputs/sz_gwas/pgc2_clozuk/
 ↪map_phase3/_m/libd_hg38_pgc2sz_snps.tsv'
}

config_feature = {
    'de_file': '../../differential_expression/_m/genes/diffExpr_szVctl_full.
 ↪txt',
    'residual_expression_file': '../../differential_expression/_m/genes/
 ↪residualized_expression.tsv',
    'fastqtl_output_file': '../../eqtl/caudate/summary_table/_m/
 ↪Brainseq_LIBD_caudate_4features.signifpairs.txt.gz',
}
```

```
feature = "genes"
```

[3]:
```
@functools.lru_cache()
def feature_map(feature):
    return {"genes": "Gene", "transcripts": "Transcript",
            "exons": "Exon", "junctions": "Junction"}[feature]


@functools.lru_cache()
def get_de_df():
    """
    Load DE analysis
    """
    return pd.read_csv(config_feature['de_file'], sep='\t', index_col=0)


@functools.lru_cache()
def get_eqtl_df():
    eqtl_df = pd.read_csv(config_feature['fastqtl_output_file'], sep='\t')
    return eqtl_df[(eqtl_df["Type"] == feature_map(feature))]


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


@functools.lru_cache()
def get_integration_df():
    return get_gwas_snps().merge(get_eqtl_df(), left_on='our_snp_id',␣
 ↪right_on='variant_id',
                                 suffixes=['_PGC2', '_eQTL'])\
                        .merge(get_de_df(), left_on='gene_id',␣
 ↪right_index=True)


@functools.lru_cache()
def get_residual_expression_df():
    return pd.read_csv(config_feature['residual_expression_file'],
                       sep='\t', index_col=0).transpose()


@functools.lru_cache()
def get_pheno_df():
    return pd.read_csv(config['phenotype_file'], index_col=0)
```

[4]:
```
def agree_direction(row):
    return [-1, 1][row['pgc2_a1_same_as_our_counted']] * np.sign(row['OR'] - 1)␣
 ↪* np.sign(row['slope']) * np.sign(row['t'])


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))))


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

[5]:
```
@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_plink_tuple():
    '''
    Usage: (bim, fam, bed) = get_plink_tuple()
    '''
    return read_plink(config['plink_file_prefix'])


@functools.lru_cache()
def subset_bed():
    """
    This subsets the bed and bim file and returns the new subsetted
    data with shared brain_ids.
```

```
    This is to speed up accessing the bed file.
    """
    (bim, fam, bed) = get_plink_tuple()
    brain_ids = list(set(get_expression_and_pheno_df()['BrNum']).
↪intersection(set(fam['fid'])))
    fam_pos = list(fam[(fam["fid"].isin(brain_ids))].
↪drop_duplicates(subset="fid").loc[:, 'i'])
    unique_snps = get_eqtl_df().variant_id.unique()
    snp_info = bim[(bim["snp"].isin(unique_snps))].copy()
    snp_pos = list(snp_info.loc[:, "i"])
    new_bed = bed[snp_pos].compute()[:,fam_pos]
    new_bim = bim[(bim["i"].isin(snp_pos))].reset_index(drop=True)
    new_bim['ii'] = new_bim.index
    return new_bed, new_bim, brain_ids


@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
    '''
    bed, bim, brain_ids = subset_bed()
    snp_info = bim[bim['snp']==snp_id]
    snp_pos = snp_info.iloc[0]['ii']
    dfsnp = pd.DataFrame(bed[[snp_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'])
    # the 2 - in next line is to workaround a possible bug in pandas_plink? a1␣
↪and a0 inverted
    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


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


@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
```

[6]:
```
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 save_plot(p, fn):
    for ext in ['png', 'pdf', 'svg']:
        p.save(fn + '.' + ext)


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_on='BrNum', right_index=True)


def simple_snp_expression_pheno_plot_impl(snp_id, gene_id, snp_df_func,␣
↪pheno_var):
    df = get_snp_gene_pheno_df(snp_id, gene_id, snp_df_func)
    df['Dx'] = df.Dx.astype('category').cat.rename_categories({'Control':␣
↪'CTL', 'Schizo': 'SZ'})
    y0 = df[gene_id].quantile(.01) - 0.26
    y1 = df[gene_id].quantile(.99) + 0.26
    pjd = position_jitterdodge(jitter_width=0.27)
    p = ggplot(df, aes(x=snp_id, y=gene_id, fill=pheno_var)) \
    + geom_boxplot(alpha=0.4, outlier_alpha=0) \
    + geom_jitter(position=pjd, stroke=0, alpha=0.6) + ylim(y0, y1) \
    + labs(y='Residualized expression', fill='Diagnosis') \
    + theme_bw(base_size=20)\
    + theme(legend_title=element_text(face='bold'),
            panel_grid_major=element_blank(),
```

```
            panel_grid_minor=element_blank())
    return p


def simple_gwas_ordered_snp_expression_pheno_plot(snp_id, gene_id, pheno_var):
    return simple_snp_expression_pheno_plot_impl(snp_id, gene_id,␣
 ↪get_gwas_ordered_snp_df, pheno_var)


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):
    r = get_eqtl_df()[(get_eqtl_df()['variant_id']==snp_id) &
                      (get_eqtl_df()['gene_id']==gene_id)]
    assert len(r)==1
    return 'eQTL nominal p-value: %.1e' % r.iloc[0]['pval_nominal']


def de_annotation(gene_id):
    g = get_de_df()[(get_de_df()['gencodeID'] == gene_id)]
    return 'DE adj.P.Val: %.3f' % g.iloc[0]['adj.P.Val']


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


def gwas_annotated_eqtl_pheno_plot(snp_id, gene_id, pheno_var):
    p = simple_gwas_ordered_snp_expression_pheno_plot(snp_id, gene_id,␣
 ↪pheno_var)
    gene_symbol, gene_description = get_gene_symbol(gene_id)
    title ="\n".join([gene_symbol, gene_id,
                      gwas_annotation(snp_id),
                      risk_allele_annotation(snp_id),
                      eqtl_annotation(snp_id, gene_id),
                      de_annotation(gene_id)])

    p += ggtitle(title)
    return p
```

1.2 Genes

[7]:
```
try:
    os.makedirs(feature)
except OSError as e:
    if e.errno != errno.EEXIST:
        raise
```

1.2.1 Enrichment

Integrate DEG with PGC2+CLOZUK SNPs

[8]:
```
dft = get_integration_df()
dft.shape
```

```
/home/jbenja13/.local/lib/python3.9/site-packages/numpy/lib/arraysetops.py:583:
FutureWarning: elementwise comparison failed; returning scalar instead, but in
the future will perform elementwise comparison
```

[8]: (1616724, 52)

[9]:
```
dft['agree_direction'] = dft.apply(agree_direction, axis=1)
agreement = {1: 'Yes', -1: 'No', 0: 0}
dft.agree_direction = [agreement[item] for item in dft['agree_direction']]
## Enrichment test
table =  [[np.sum((dft['P']<5e-8) & ((dft['adj.P.Val']<.05))),
           np.sum((dft['P']<5e-8) & ((dft['adj.P.Val']>=.05)))],
          [np.sum((dft['P']>=5e-8) & ((dft['adj.P.Val']<.05))),
           np.sum((dft['P']>=5e-8) & ((dft['adj.P.Val']>=.05)))]]
print(table)
fisher_exact(table)
```

```
[[3020, 33767], [168656, 1411281]]
```

[9]: (0.74838687753065, 1.3150145933372704e-55)

[10]:
```
dft1 = dft[(dft['P']<5e-8) & ((dft['adj.P.Val']<.05))]
df = dft1.groupby('agree_direction').size().reset_index()
df
```

[10]:

	agree_direction	0
0	No	688
1	Yes	2332

[11]:
```
binom_test(df[0].iloc[1], df[0].sum())
```

[11]: 2.4968994585676953e-207

[12]:
```
dft2 = dft[(dft['P']<=5e-8) & (dft['adj.P.Val'] < 0.05)].copy()
dft2.groupby("gene_id").first().reset_index().shape
```

[12]:
```
(35, 53)
```

[13]:
```
dft2['risk_allele'] = dft2['our_snp_id'].apply(get_risk_allele)
```

[14]:
```
direction = {-1: 'Down', 1: 'Up'}
boolean_conv = {True: 1, False: -1}
dft2.pgc2_a1_same_as_our_counted = [boolean_conv[item] for item in␣
↪dft2['pgc2_a1_same_as_our_counted']]
dft2['eqtl_gwas_dir'] = [direction[item] for item in np.
↪sign(dft2['pgc2_a1_same_as_our_counted']) * np.sign(dft2['slope']) * np.
↪sign(dft2['OR'] - 1)]
dft2['de_dir'] = [direction[item] for item in np.sign(dft2['t'])]
dft2['eqtl_slope'] = np.sign(dft2['pgc2_a1_same_as_our_counted']) * np.
↪sign(dft2['OR'] - 1) * dft2['slope']
dft3 = dft2[['gene_id', 'Symbol', 'variant_id', 'A1', 'A2', 'risk_allele', 'OR',
             'P', 'pval_nominal', 'adj.P.Val', 'logFC', 't', 'eqtl_slope',
             'de_dir', 'eqtl_gwas_dir', 'agree_direction']]
dft3['Symbol'].fillna(dft3['gene_id'], inplace=True)
dft3.to_csv('%s/integration_by_symbol.txt' % feature, sep='\t', index=False)
```

```
/home/jbenja13/.local/lib/python3.9/site-packages/pandas/core/series.py:4463:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

[15]:
```
#dft3 = pd.read_csv("../_m/genes/integration_by_symbol.txt", sep='\t')
#dft3['Symbol'].fillna(dft3['gene_id'], inplace=True)
df2 = dft3.groupby(['gene_id']).first().reset_index().sort_values('P')
df2.groupby(['agree_direction']).size()
```

[15]:
```
agree_direction
No     13
Yes    22
dtype: int64
```

[16]:
```
df2.set_index('Symbol').rename(columns={'t': 'de_t', 'P': 'GWAS_P',␣
↪'pval_nominal': 'eQTL_pvalue',
                                        'adj.P.Val': 'de_FDR'})
```

[16]:

Symbol	gene_id	variant_id	A1	A2	\
ZSCAN9	ENSG00000137185.11	chr6:27731058:C:T	C	T	
HCG4	ENSG00000176998.4	chr6:29298706:G:A	G	A	
BRD2	ENSG00000204256.12	chr6:32669525:G:A	G	A	
FLOT1	ENSG00000137312.14	chr6:30406434:A:G	A	G	
HCG11	ENSG00000228223.2	chr6:26466161:G:A	G	A	

BAG6	ENSG00000204463.12	chr6:31222761:G:A	G	A	
NELFE	ENSG00000204356.12	chr6:31868665:A:G	A	G	
ZNF204P	ENSG00000204789.4	chr6:27343879:A:G	A	G	
NGEF	ENSG00000066248.14	chr2:232843683:G:A	G	A	
PRRC2A	ENSG00000204469.12	chr6:31379292:C:T	C	T	
PLCH2	ENSG00000149527.17	chr1:2440958:A:G	A	G	
IP6K3	ENSG00000161896.11	chr6:33541236:A:G	A	G	
DNAJC19	ENSG00000205981.6	chr3:181087982:C:G	C	G	
LINC01470	ENSG00000249484.8	chr5:152727548:A:G	A	G	
CNNM2	ENSG00000148842.17	chr10:102825368:C:A	C	A	
ZNF391	ENSG00000124613.8	chr6:26990749:T:C	T	C	
PLPP5	ENSG00000147535.16	chr8:38263203:T:C	T	C	
PTN	ENSG00000105894.11	chr7:137363413:T:C	T	C	
TRMT61A	ENSG00000166166.12	chr14:103541799:CAGG:C	CAGG	C	
PCCB	ENSG00000114054.13	chr3:136088767:G:T	G	T	
PPTC7	ENSG00000196850.5	chr12:110126854:A:C	A	C	
TSNARE1	ENSG00000171045.14	chr8:142261784:C:T	C	T	
HIRIP3	ENSG00000149929.15	chr16:29959536:C:G	C	G	
PPM1M	ENSG00000164088.17	chr3:52253452:G:T	G	T	
VRK2	ENSG00000028116.16	chr2:57874972:T:A	T	A	
ZC3H7B	ENSG00000100403.11	chr22:41357599:G:A	G	A	
REEP2	ENSG00000132563.15	chr5:138362610:C:T	C	T	
KDM3B	ENSG00000120733.13	chr5:138362610:C:T	C	T	
SREBF2	ENSG00000198911.11	chr22:41830391:C:T	C	T	
BNIP3L	ENSG00000104765.15	chr8:26368096:G:A	G	A	
ENSG00000228944.1	ENSG00000228944.1	chr7:24695385:G:C	G	C	
PHF1	ENSG00000112511.17	chr6:33343202:G:A	G	A	
C4A	ENSG00000244731.7	chr6:31482137:T:C	T	C	
ZNF14	ENSG00000105708.8	chr19:19623068:T:C	T	C	
ENSG00000253553.5	ENSG00000253553.5	chr8:88271460:A:G	A	G	

Symbol	risk_allele	OR	GWAS_P	eQTL_pvalue	\
ZSCAN9	C	1.26220	1.210000e-39	8.946610e-04	
HCG4	G	1.26360	2.500000e-39	4.443110e-05	
BRD2	G	1.21970	6.820000e-30	1.131810e-05	
FLOT1	A	1.23630	2.800000e-27	6.289760e-04	
HCG11	A	0.91432	1.020000e-14	1.374100e-05	
BAG6	A	0.92265	6.790000e-14	8.199240e-05	
NELFE	A	1.07930	1.660000e-13	7.305820e-04	
ZNF204P	G	0.93306	4.310000e-13	1.058040e-03	
NGEF	A	0.91758	7.130000e-13	2.415450e-04	
PRRC2A	C	1.08610	7.930000e-12	4.021460e-04	
PLCH2	G	0.92873	4.630000e-11	2.811440e-09	
IP6K3	G	0.93912	2.350000e-10	1.762100e-04	
DNAJC19	C	1.07600	7.190000e-10	4.418510e-04	
LINC01470	A	1.06910	8.730000e-10	5.630250e-14	

CNNM2	C	1.06040	1.120000e-09	2.413790e-11
ZNF391	C	0.94335	1.550000e-09	1.247880e-03
PLPP5	T	1.06970	2.490000e-09	3.233620e-04
PTN	C	0.94234	2.770000e-09	2.497190e-04
TRMT61A	C	0.93885	4.218000e-09	1.193570e-04
PCCB	T	0.94493	5.050000e-09	2.185890e-14
PPTC7	A	1.06290	5.810000e-09	7.078260e-04
TSNARE1	C	1.07400	1.150000e-08	4.723740e-05
HIRIP3	C	1.05670	1.190000e-08	6.256180e-04
PPM1M	G	1.05600	1.350000e-08	6.597780e-04
VRK2	A	0.91021	1.370000e-08	6.451330e-04
ZC3H7B	A	0.94102	1.760000e-08	1.177050e-07
REEP2	C	1.05600	1.860000e-08	3.010360e-07
KDM3B	C	1.05600	1.860000e-08	1.455140e-04
SREBF2	C	1.05530	2.290000e-08	9.115040e-05
BNIP3L	A	0.93463	2.300000e-08	1.403510e-04
ENSG00000228944.1	C	0.91974	2.310000e-08	1.514450e-04
PHF1	A	0.94764	4.060000e-08	1.541960e-04
C4A	T	1.06010	4.120000e-08	1.330480e-04
ZNF14	T	1.06130	4.300000e-08	3.436760e-06
ENSG00000253553.5	A	1.06200	4.410000e-08	3.018090e-04

Symbol	de_FDR	logFC	de_t	eqtl_slope	de_dir	\
ZSCAN9	2.262693e-02	-0.077130	-3.126621	-0.307218	Down	
HCG4	1.090935e-02	0.231187	3.425733	0.374783	Up	
BRD2	1.718619e-02	-0.051562	-3.239197	-0.404029	Down	
FLOT1	1.347849e-02	-0.047316	-3.337230	-0.193544	Down	
HCG11	1.181958e-03	0.114043	4.193600	0.211170	Up	
BAG6	1.241493e-02	-0.044002	-3.373864	0.148294	Down	
NELFE	2.507792e-02	-0.046326	-3.081897	0.090267	Down	
ZNF204P	1.603175e-02	0.095817	3.267501	0.089913	Up	
NGEF	9.144005e-03	0.104229	3.493550	-0.144167	Up	
PRRC2A	1.283462e-03	-0.066383	-4.167993	-0.131287	Down	
PLCH2	1.909382e-04	-0.169308	-4.727576	-0.174917	Down	
IP6K3	2.435252e-03	-0.233018	-3.959823	0.131471	Down	
DNAJC19	2.188006e-02	0.042839	3.140810	0.098382	Up	
LINC01470	1.143030e-02	0.406227	3.407085	0.488819	Up	
CNNM2	3.899296e-03	0.053701	3.799656	-0.169331	Up	
ZNF391	2.054017e-02	0.082216	3.166580	-0.145940	Up	
PLPP5	2.838093e-02	-0.073766	-3.026111	-0.209961	Down	
PTN	7.972676e-03	-0.086066	-3.545516	0.122412	Down	
TRMT61A	4.958693e-02	-0.058611	-2.775075	-0.127611	Down	
PCCB	3.397384e-02	-0.058710	-2.946922	-0.246348	Down	
PPTC7	3.514250e-03	0.090141	3.837368	-0.119268	Up	
TSNARE1	4.121874e-03	-0.076791	-3.780711	0.177779	Down	
HIRIP3	2.093547e-04	-0.091615	-4.698442	0.139753	Down	

PPM1M	8.758407e-06	-0.136491	-5.484394	-0.116511	Down
VRK2	1.076112e-02	-0.127117	-3.431328	0.150703	Down
ZC3H7B	1.255326e-02	-0.055238	-3.367649	0.258939	Down
REEP2	6.655155e-09	0.137644	6.993625	0.145249	Up
KDM3B	1.518040e-02	-0.031591	-3.287837	-0.167676	Down
SREBF2	9.947151e-04	-0.075761	-4.243388	-0.177463	Down
BNIP3L	1.065324e-02	-0.063722	-3.435641	-0.101083	Down
ENSG00000228944.1	2.256942e-02	0.278474	3.127826	0.412318	Up
PHF1	2.281934e-02	-0.046170	-3.123441	-0.161077	Down
C4A	7.880375e-03	0.358141	3.550690	0.178003	Up
ZNF14	1.653178e-02	0.074769	3.254277	0.231031	Up
ENSG00000253553.5	2.909648e-02	0.202146	3.016220	-0.220670	Up

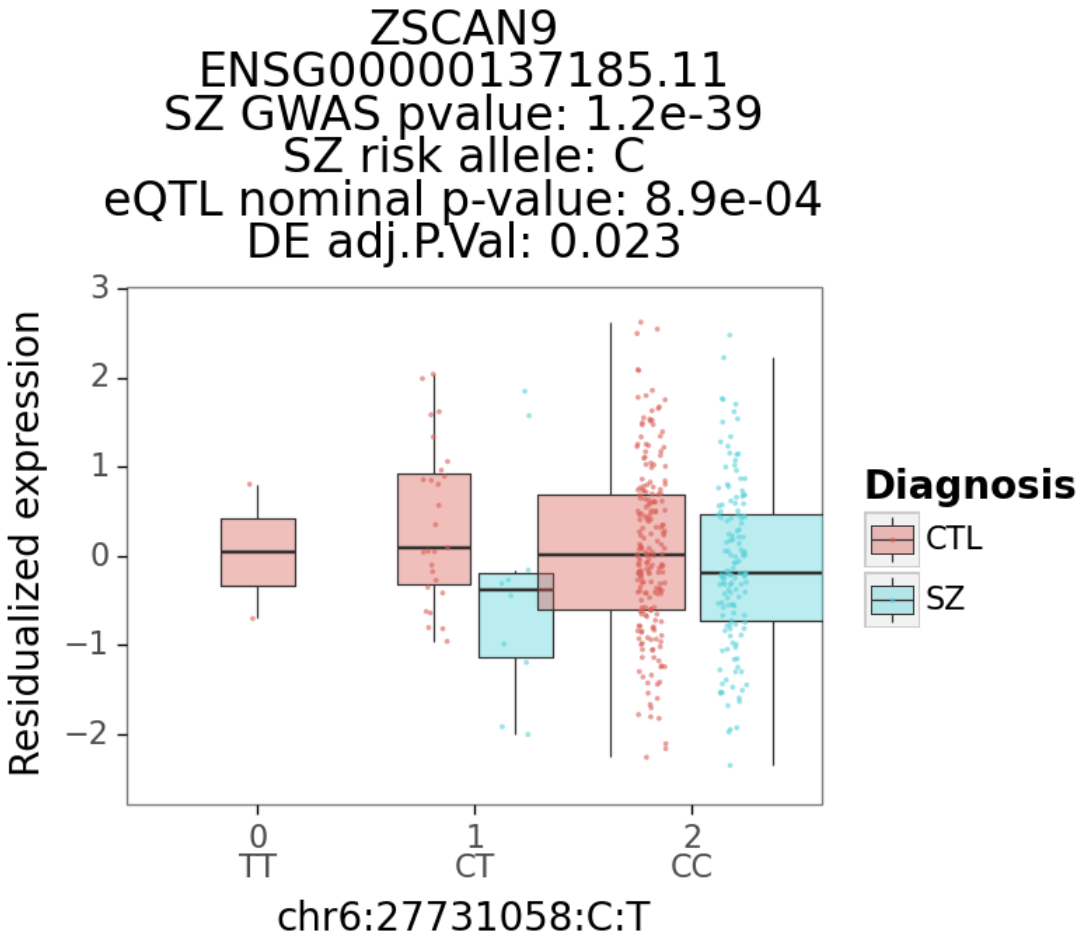
Symbol	eqtl_gwas_dir	agree_direction
ZSCAN9	Down	Yes
HCG4	Up	Yes
BRD2	Down	Yes
FLOT1	Down	Yes
HCG11	Up	Yes
BAG6	Up	No
NELFE	Up	No
ZNF204P	Up	Yes
NGEF	Down	No
PRRC2A	Down	Yes
PLCH2	Down	Yes
IP6K3	Up	No
DNAJC19	Up	Yes
LINC01470	Up	Yes
CNNM2	Down	No
ZNF391	Down	No
PLPP5	Down	Yes
PTN	Up	No
TRMT61A	Down	Yes
PCCB	Down	Yes
PPTC7	Down	No
TSNARE1	Up	No
HIRIP3	Up	No
PPM1M	Down	Yes
VRK2	Up	No
ZC3H7B	Up	No
REEP2	Up	Yes
KDM3B	Down	Yes
SREBF2	Down	Yes
BNIP3L	Down	Yes
ENSG00000228944.1	Up	Yes
PHF1	Down	Yes

C4A	Up	Yes
ZNF14	Up	Yes
ENSG00000253553.5	Down	No

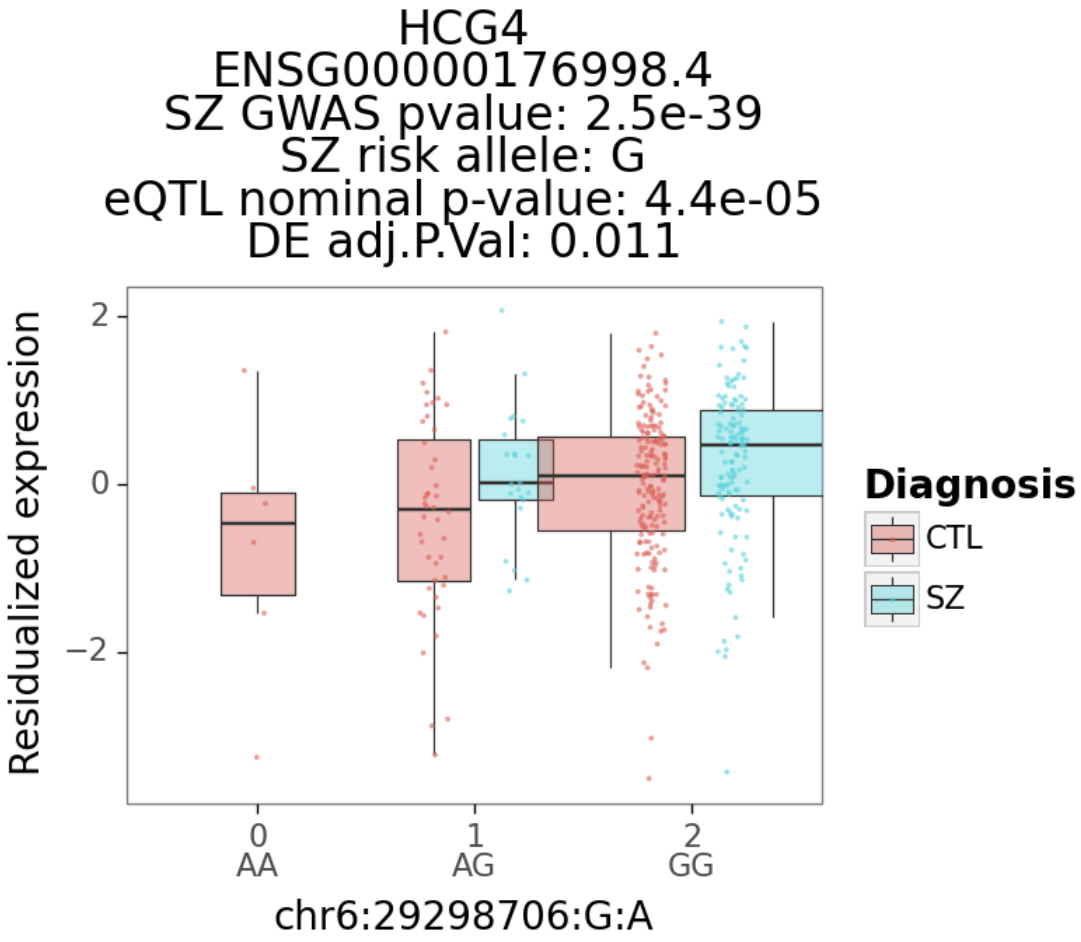
1.2.2 Plot with PGC2 risk allele

[17]:
```
for xx in range(df2.shape[0]):
    gg = gwas_annotated_eqtl_pheno_plot(df2.iloc[xx, :].variant_id, df2.
 ↪iloc[xx, :].gene_id, 'Dx')
    print(gg)
    label = '%s/eqtl_gwas_%s' % (feature, df2.iloc[xx, :].Symbol)
    save_plot(gg, label)
```

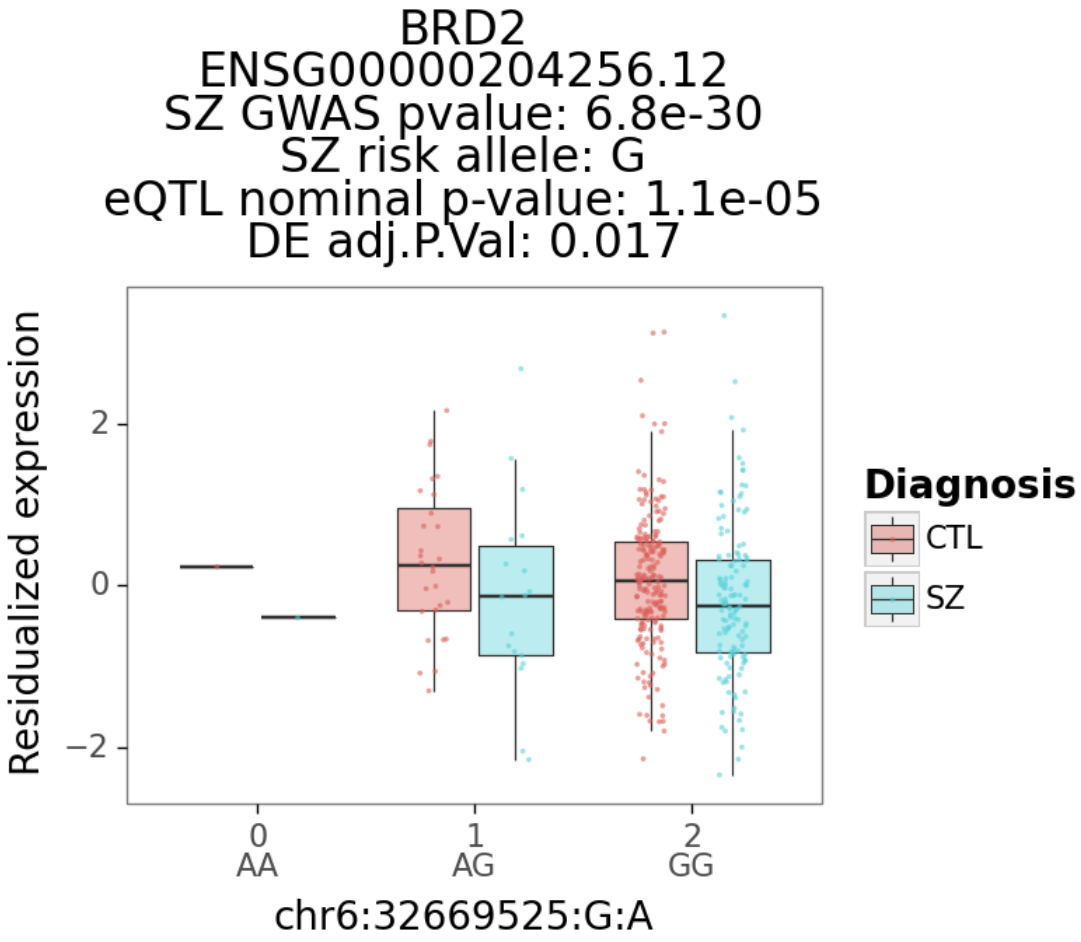
```
Mapping files: 100%|      | 3/3 [00:25<00:00,  8.53s/it]
```



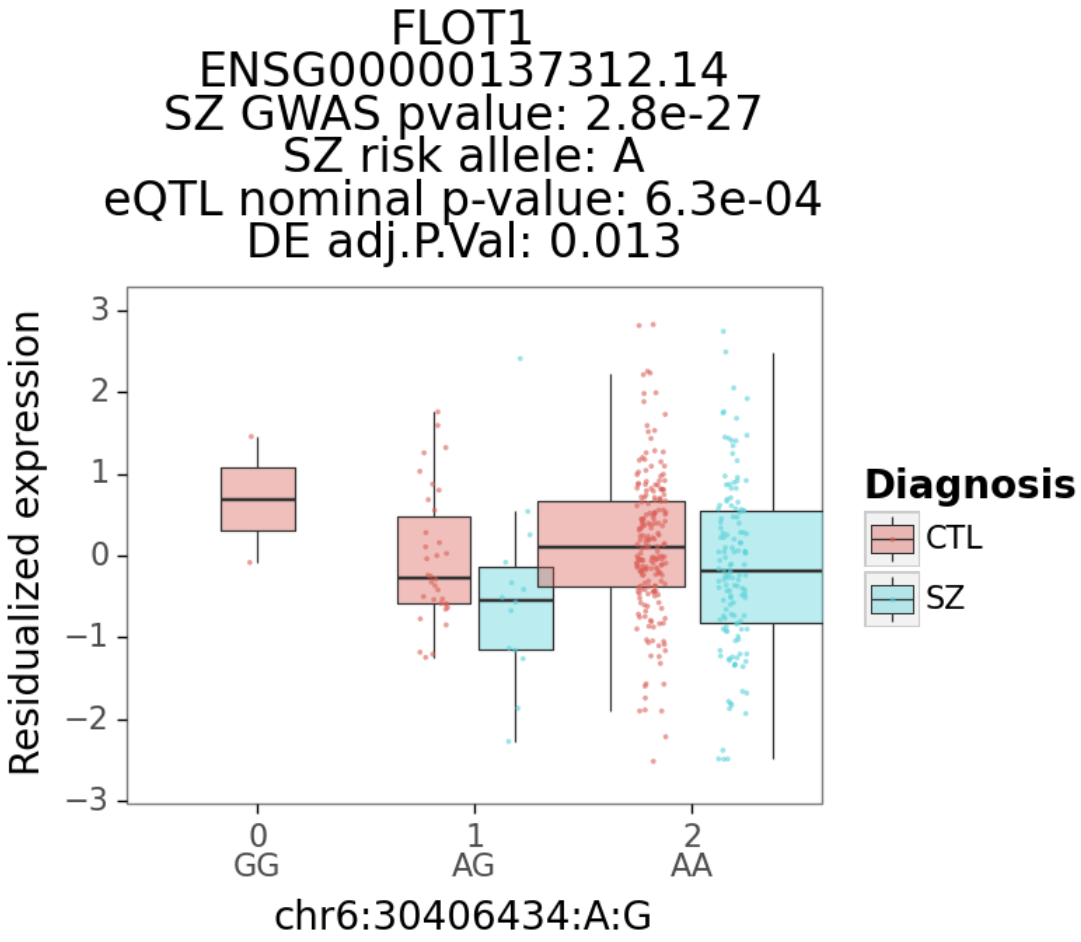

```
<ggplot: (8779647847280)>
```



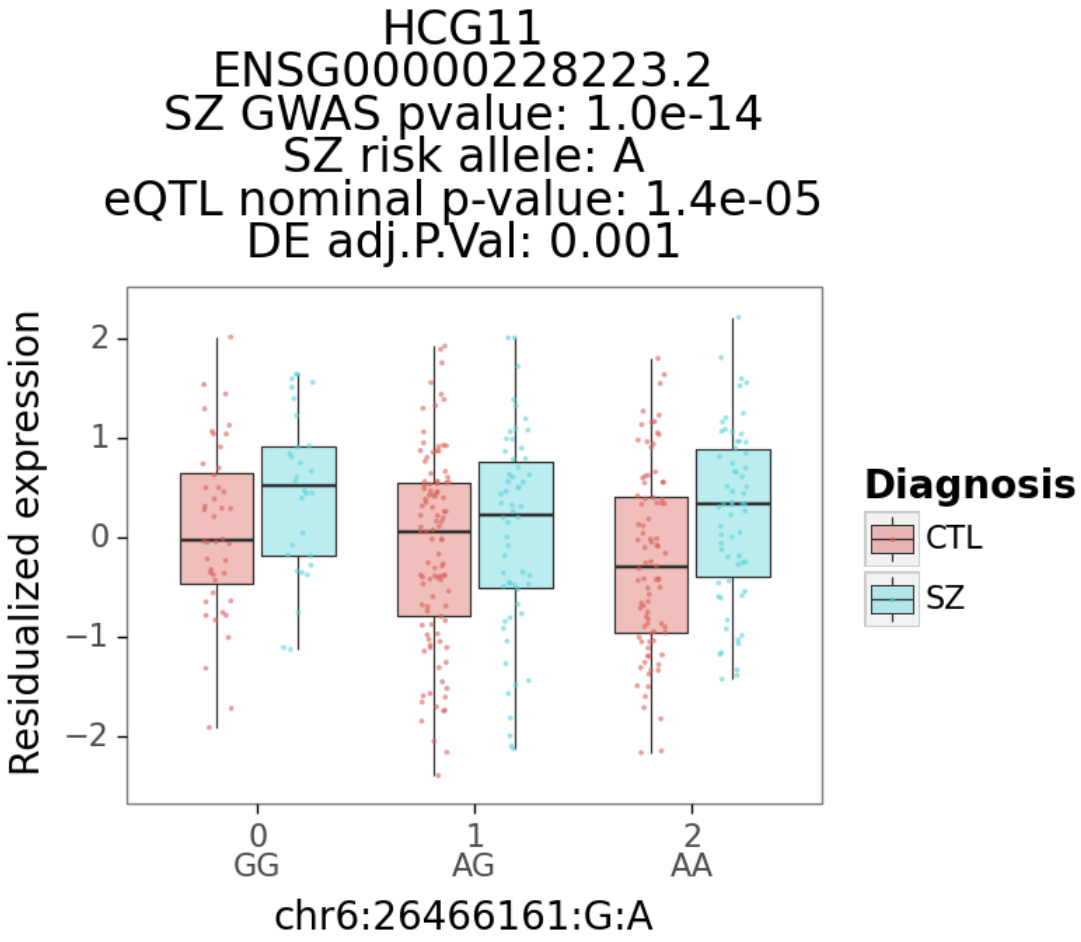

```
<ggplot: (8779648926365)>
```



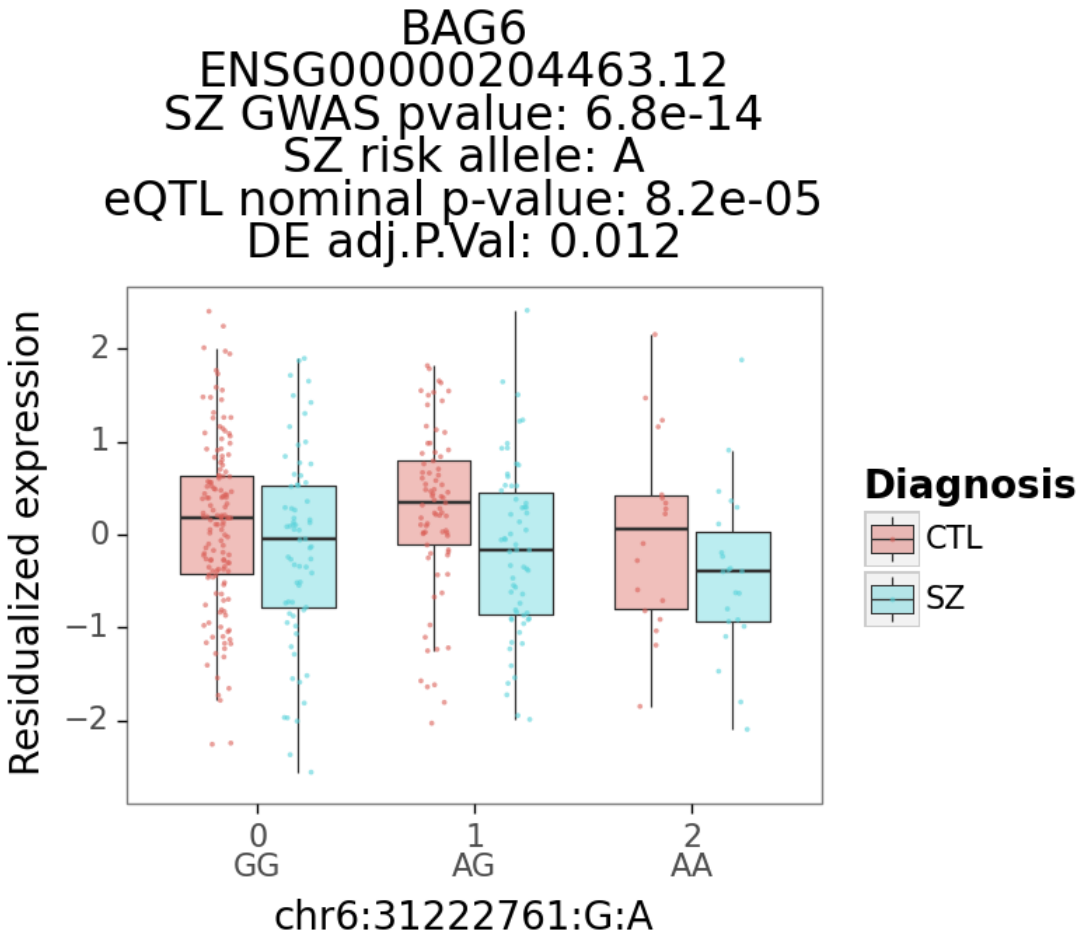

```
<ggplot: (8779649122193)>
```



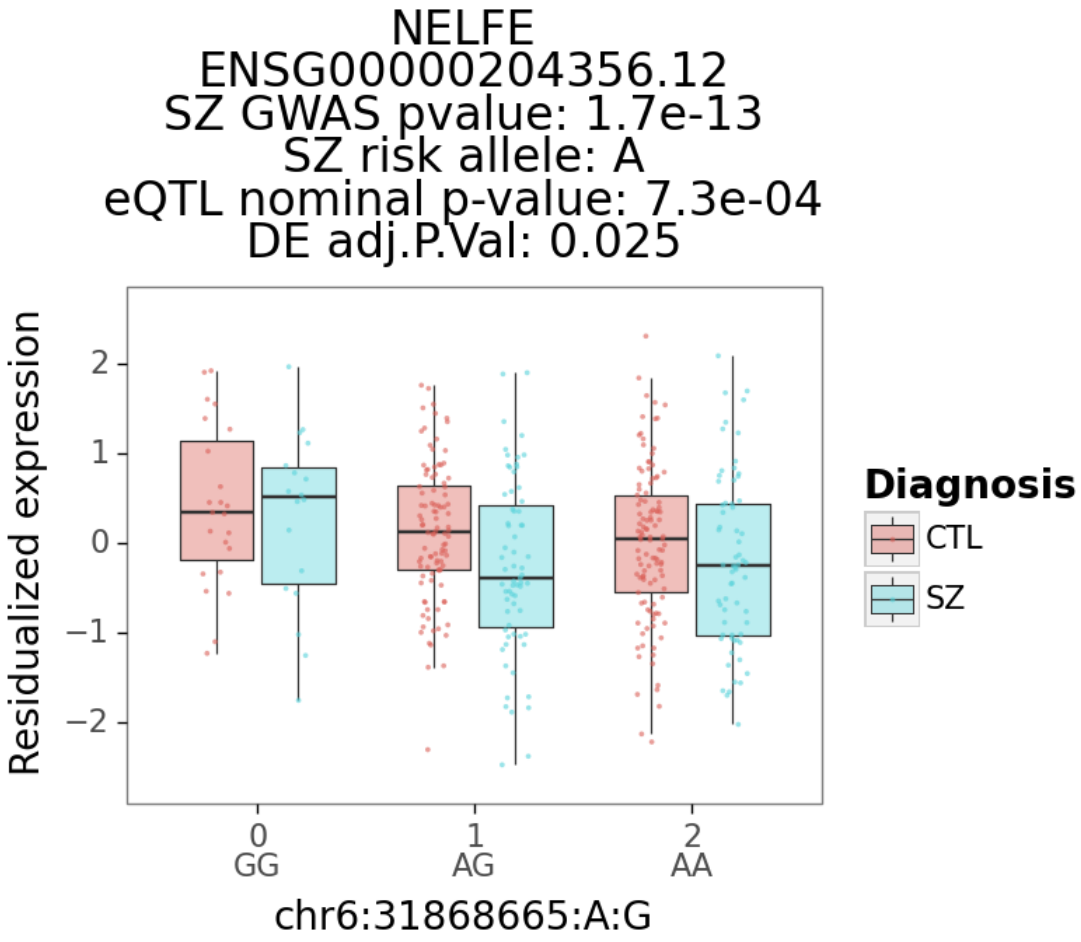

```
<ggplot: (8779673197017)>
```



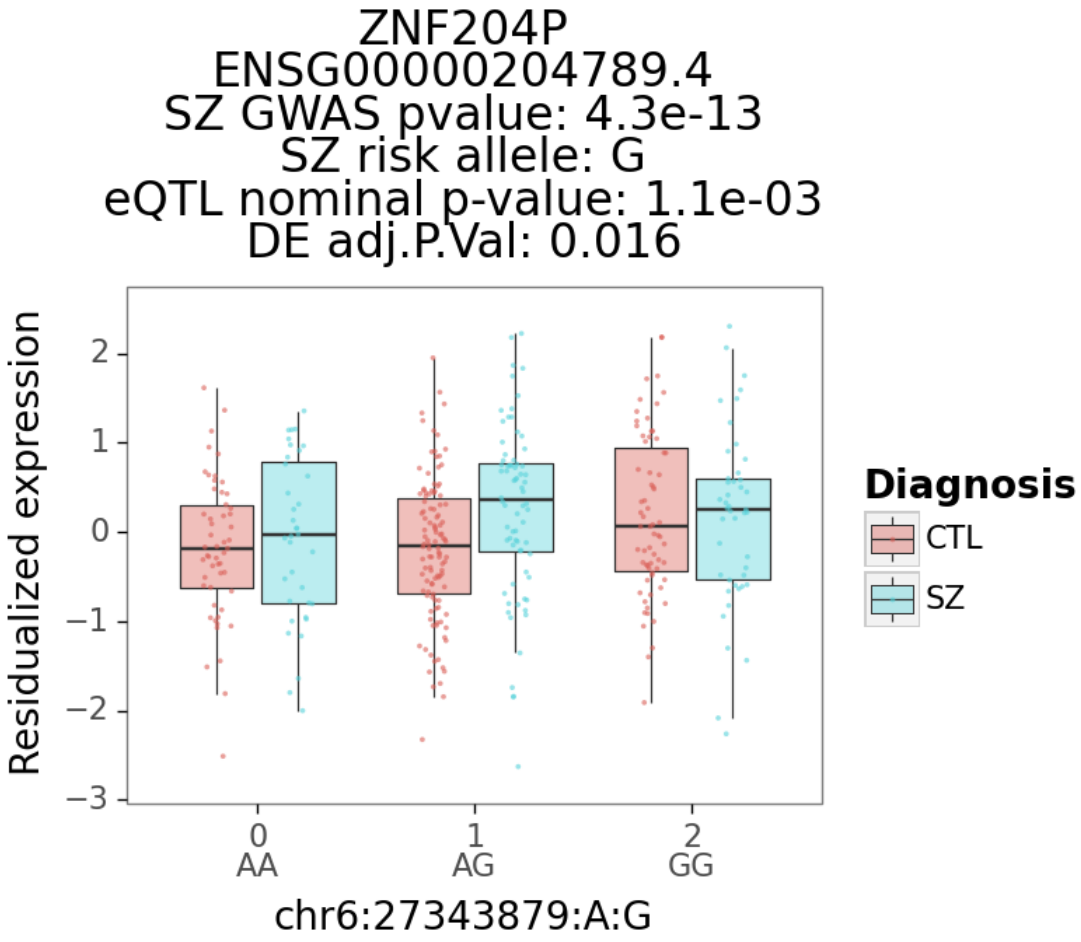

```
<ggplot: (8779673470455)>
```



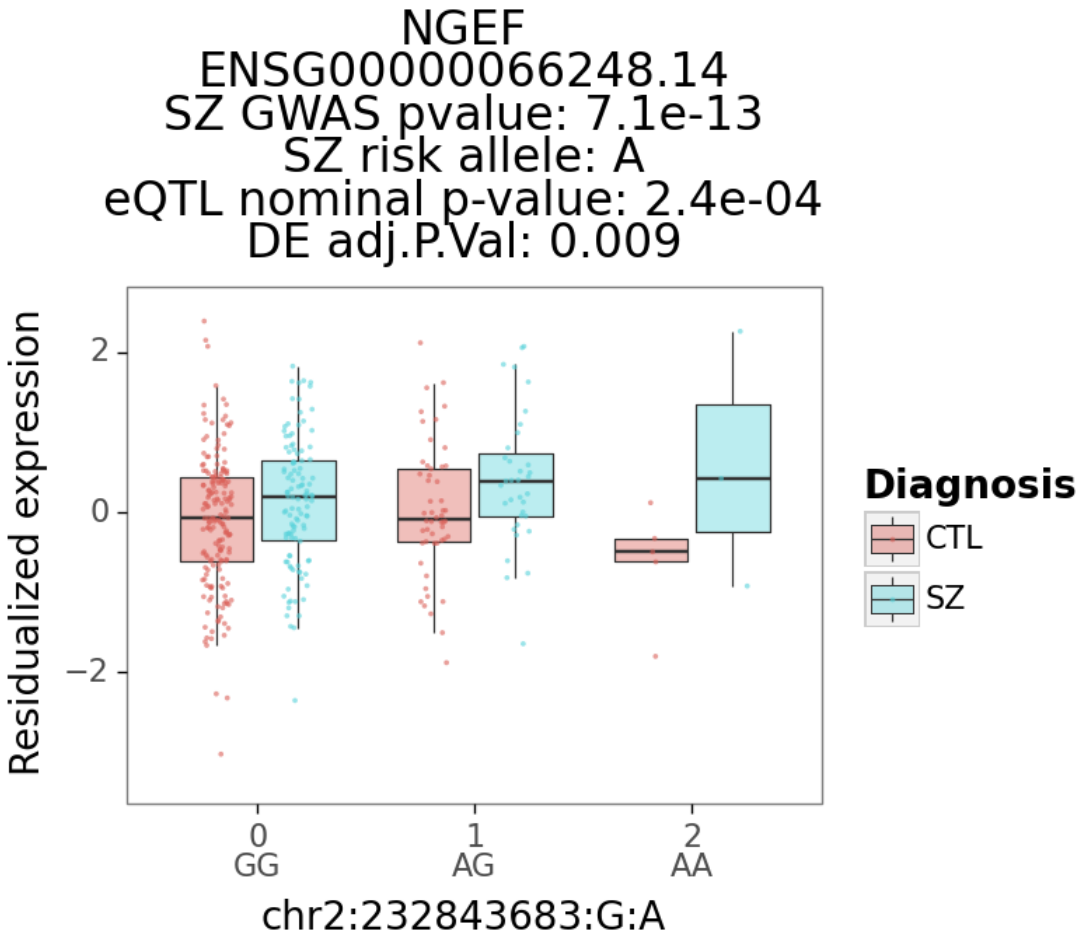

```
<ggplot: (8779647655783)>
```



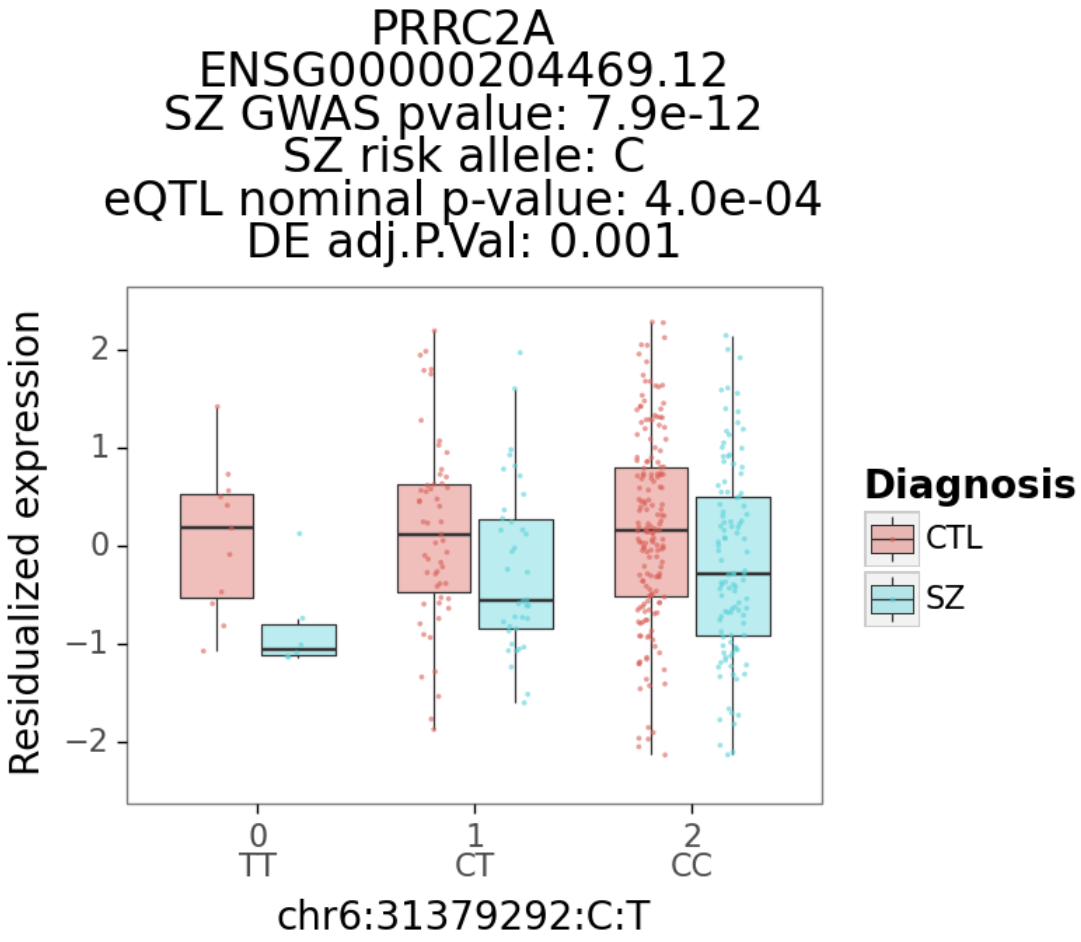

```
<ggplot: (8779616699468)>
```



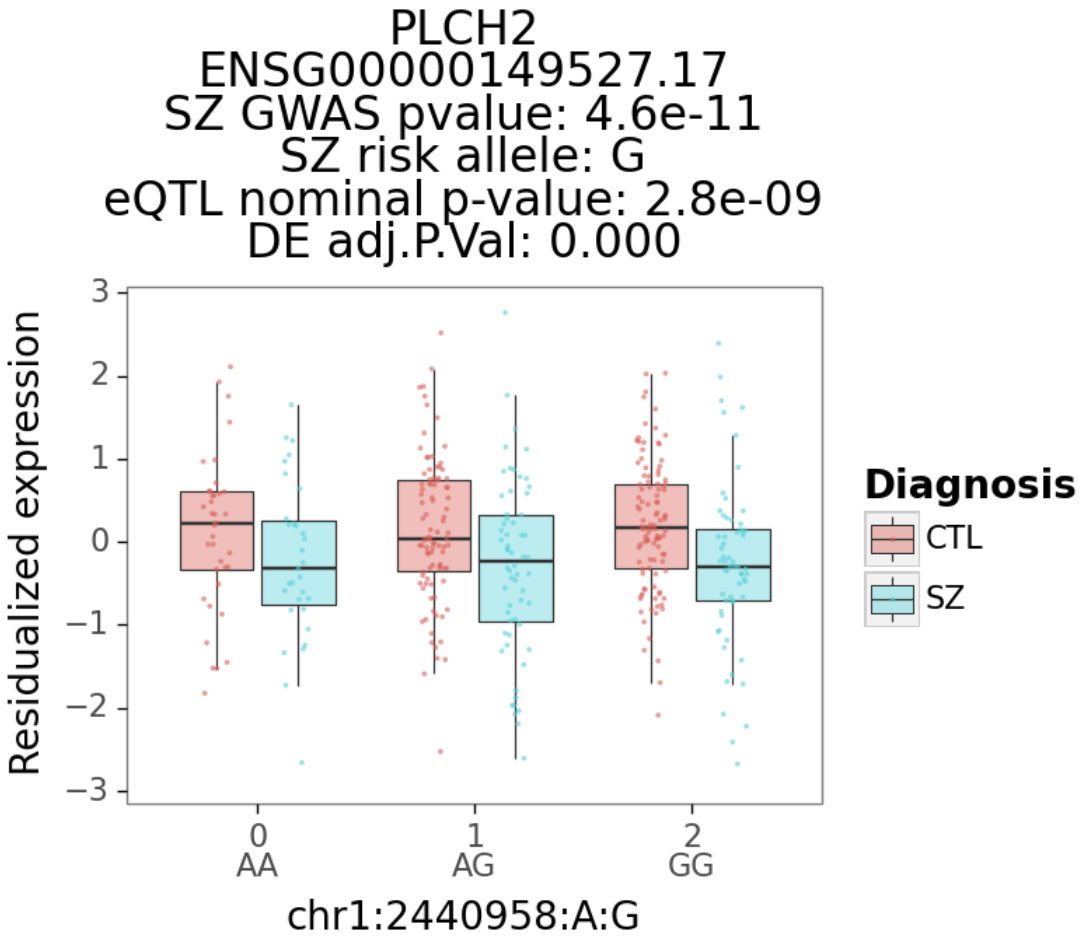

```
<ggplot: (8779647377445)>
```



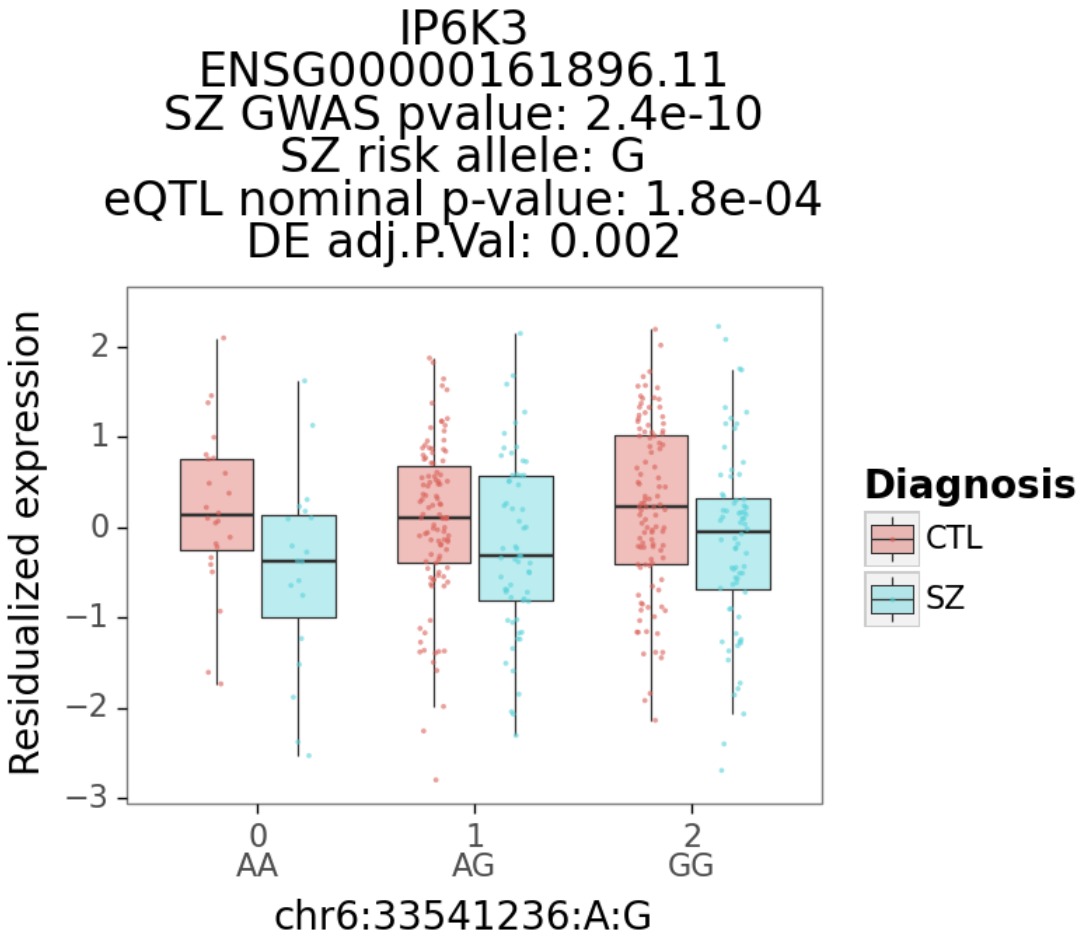

```
<ggplot: (8779674186150)>
```



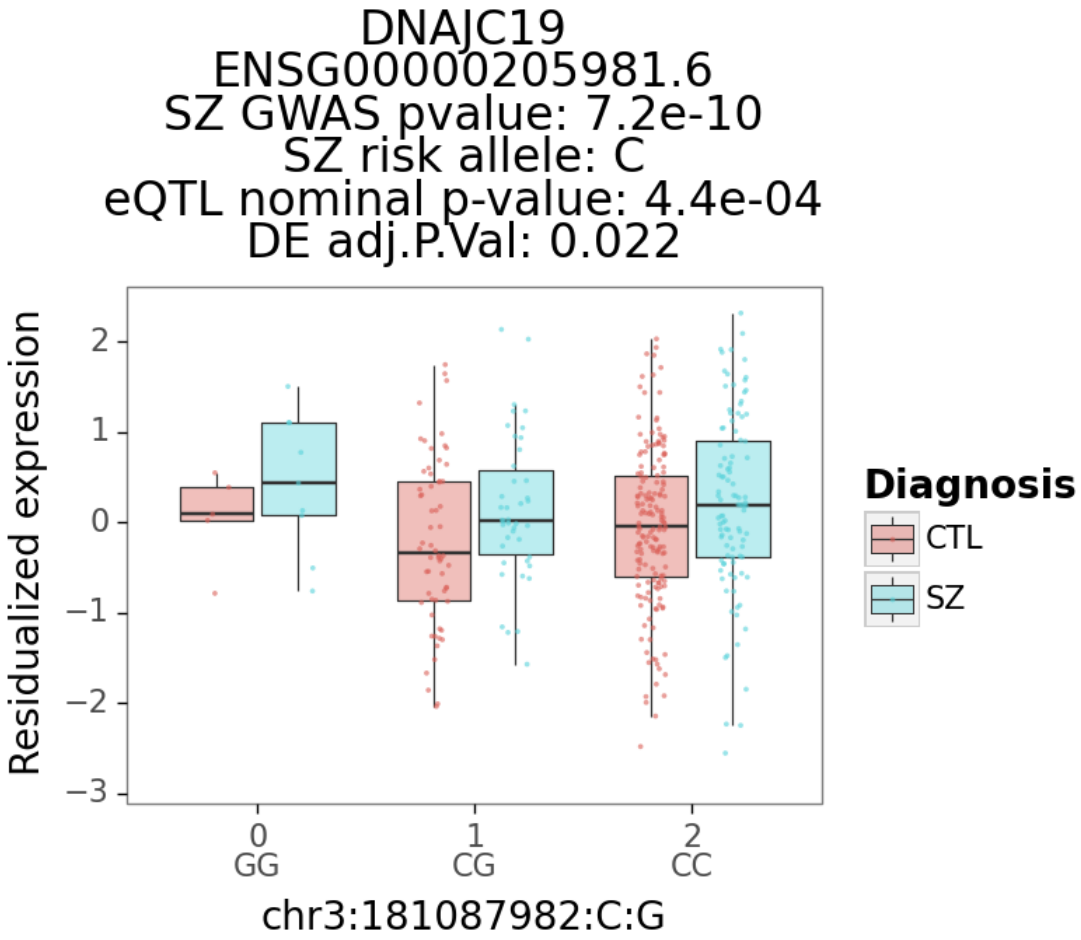

```
<ggplot: (8779648727333)>
```



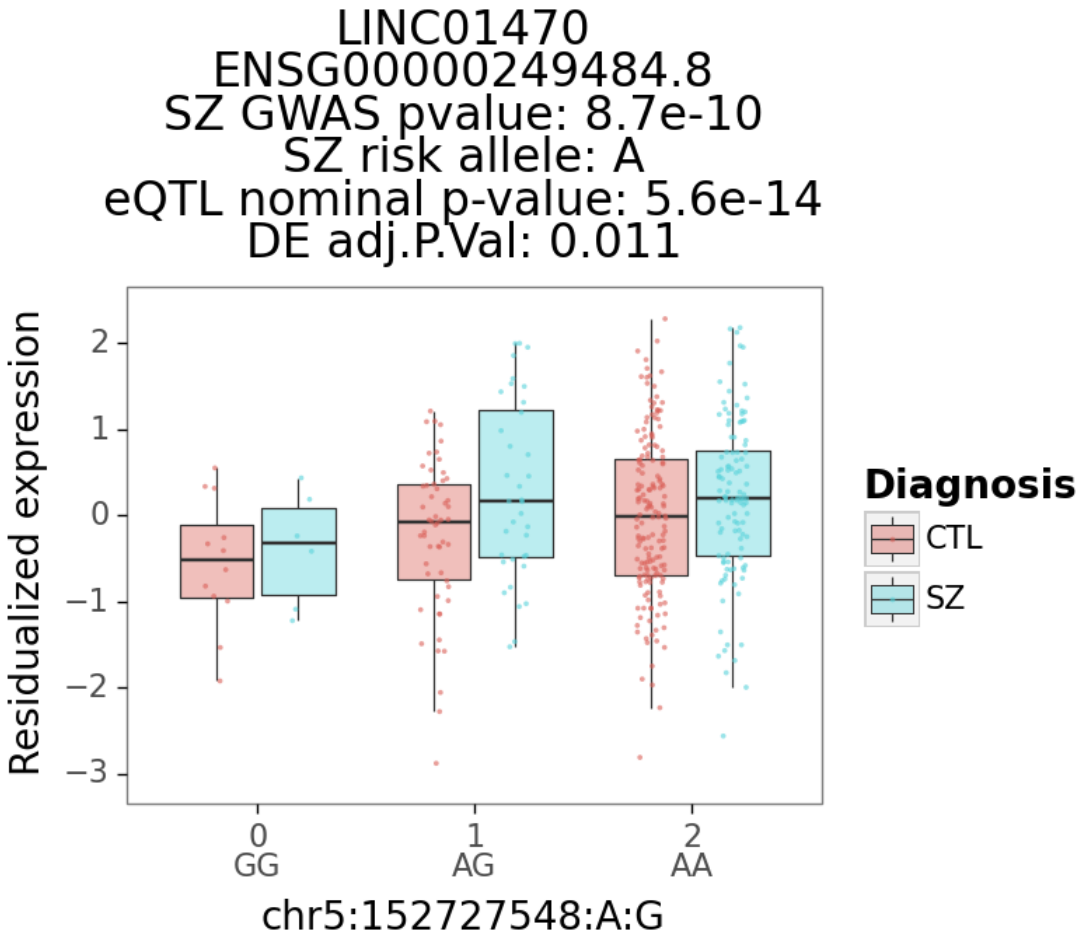

```
<ggplot: (8779648500149)>
```



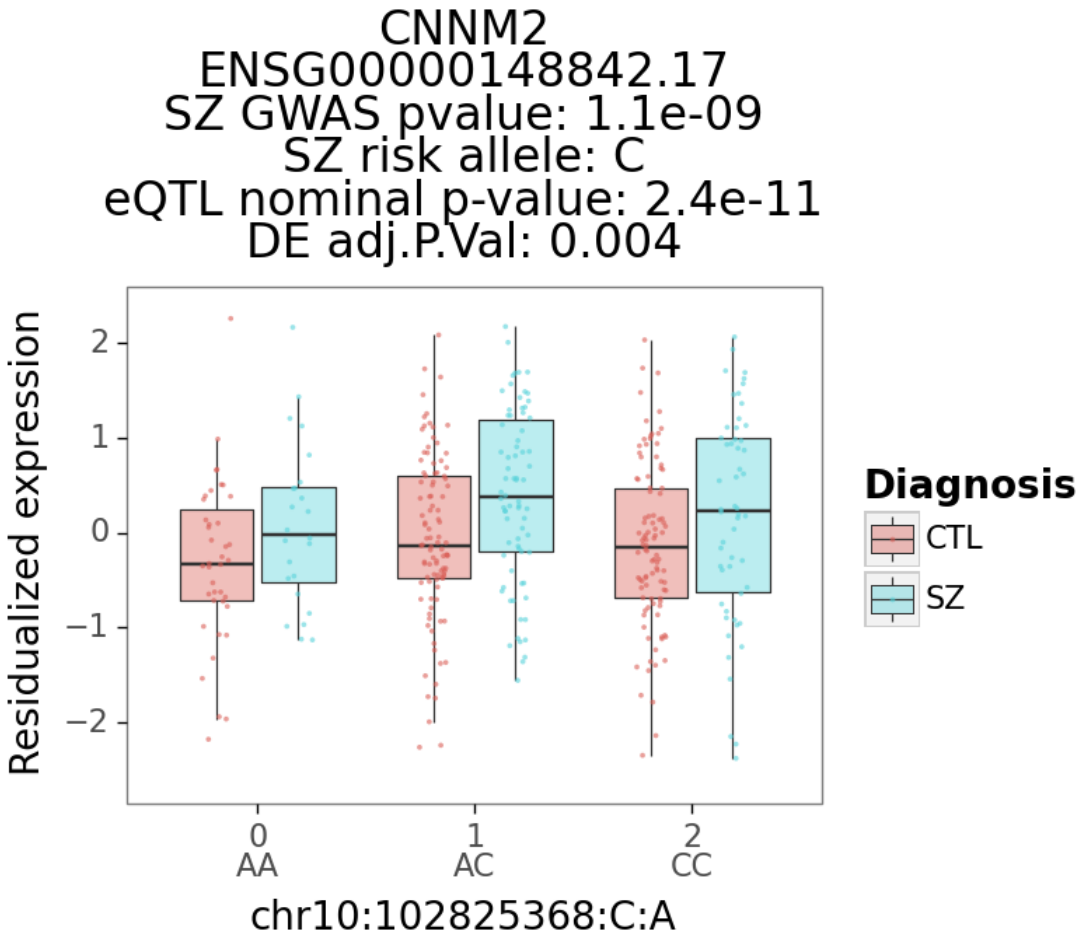

```
<ggplot: (8779648120369)>
```



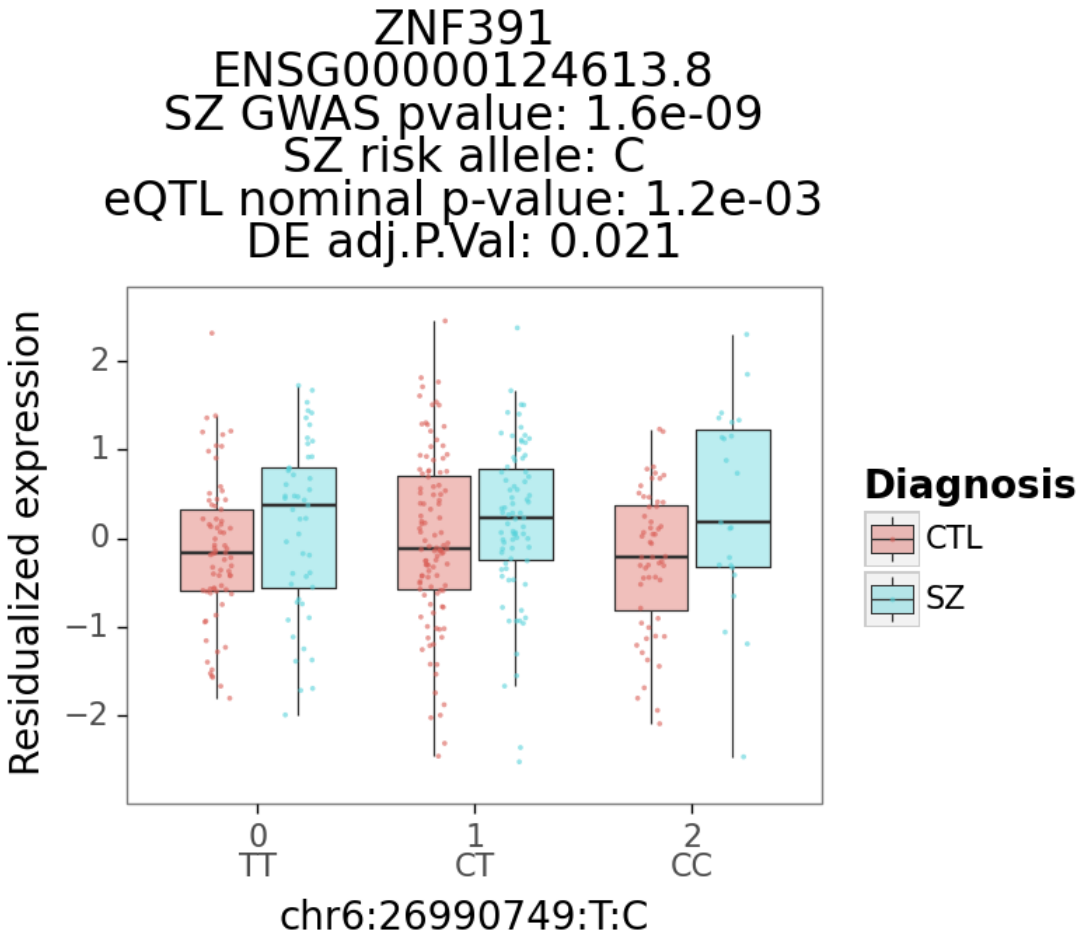

```
<ggplot: (8779615827688)>
```



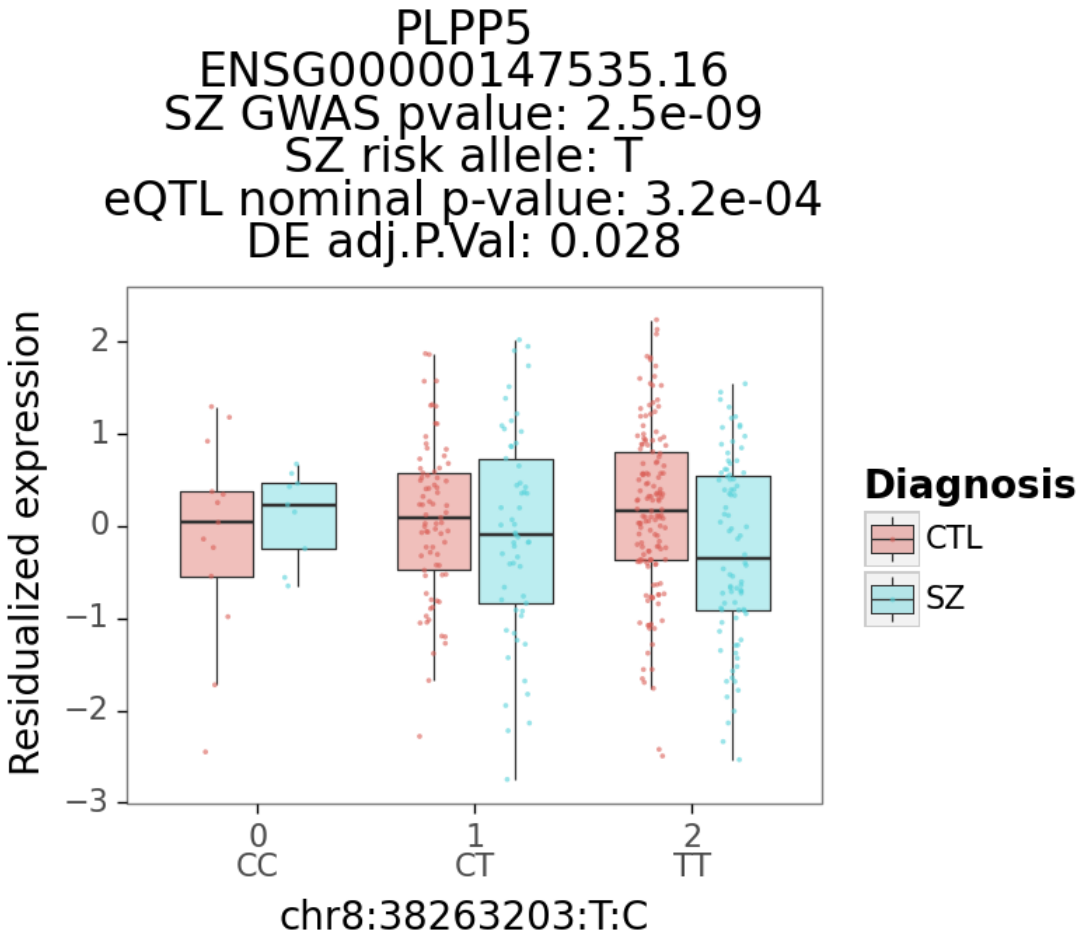

```
<ggplot: (8779648580172)>
```



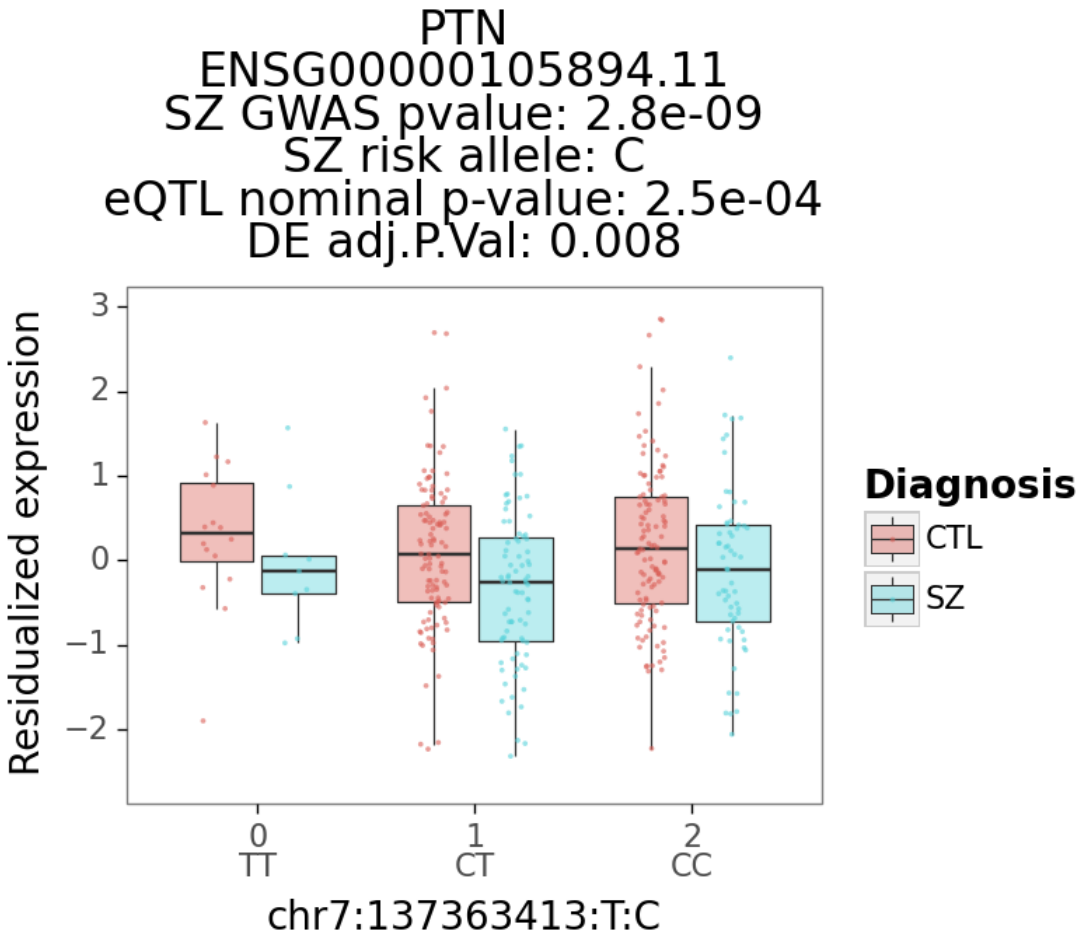

```
<ggplot: (8779674563202)>
```



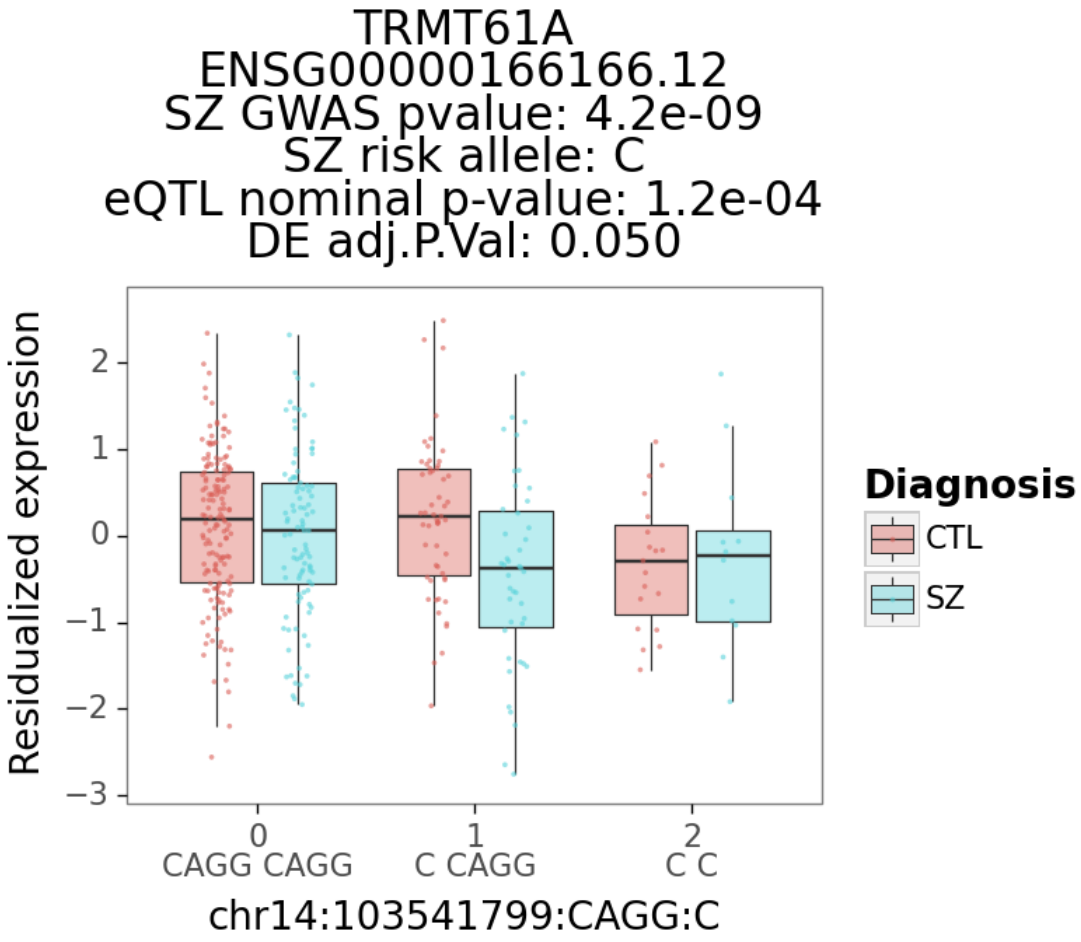

```
<ggplot: (8779647503433)>
```



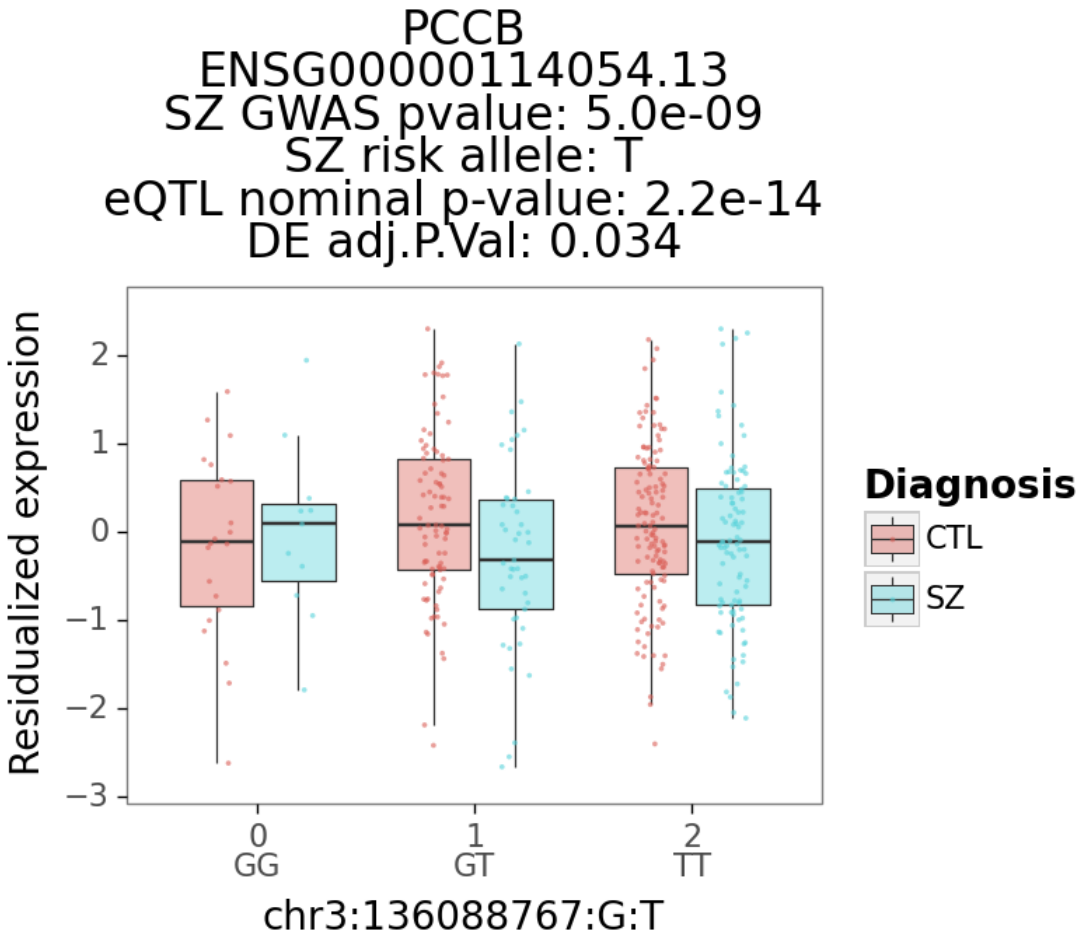

```
<ggplot: (8779674832619)>
```



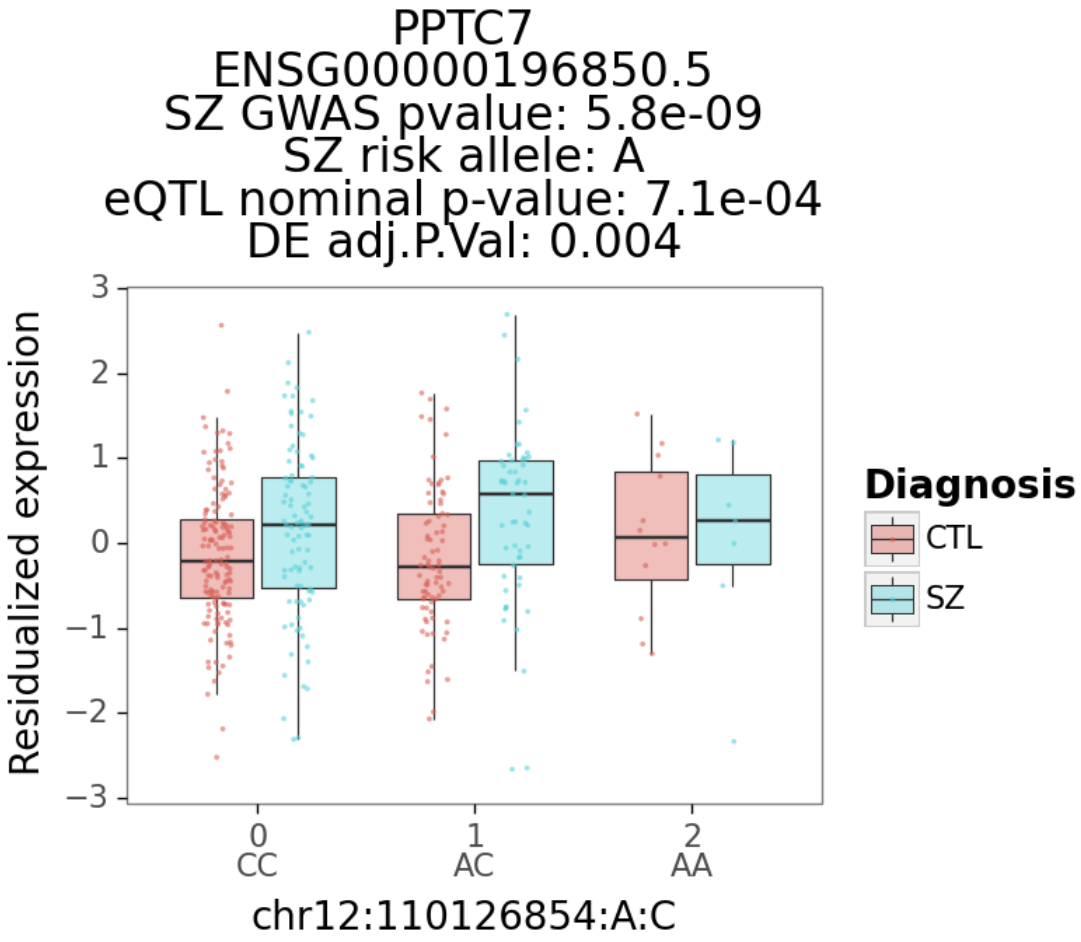

```
<ggplot: (8779673702925)>
```



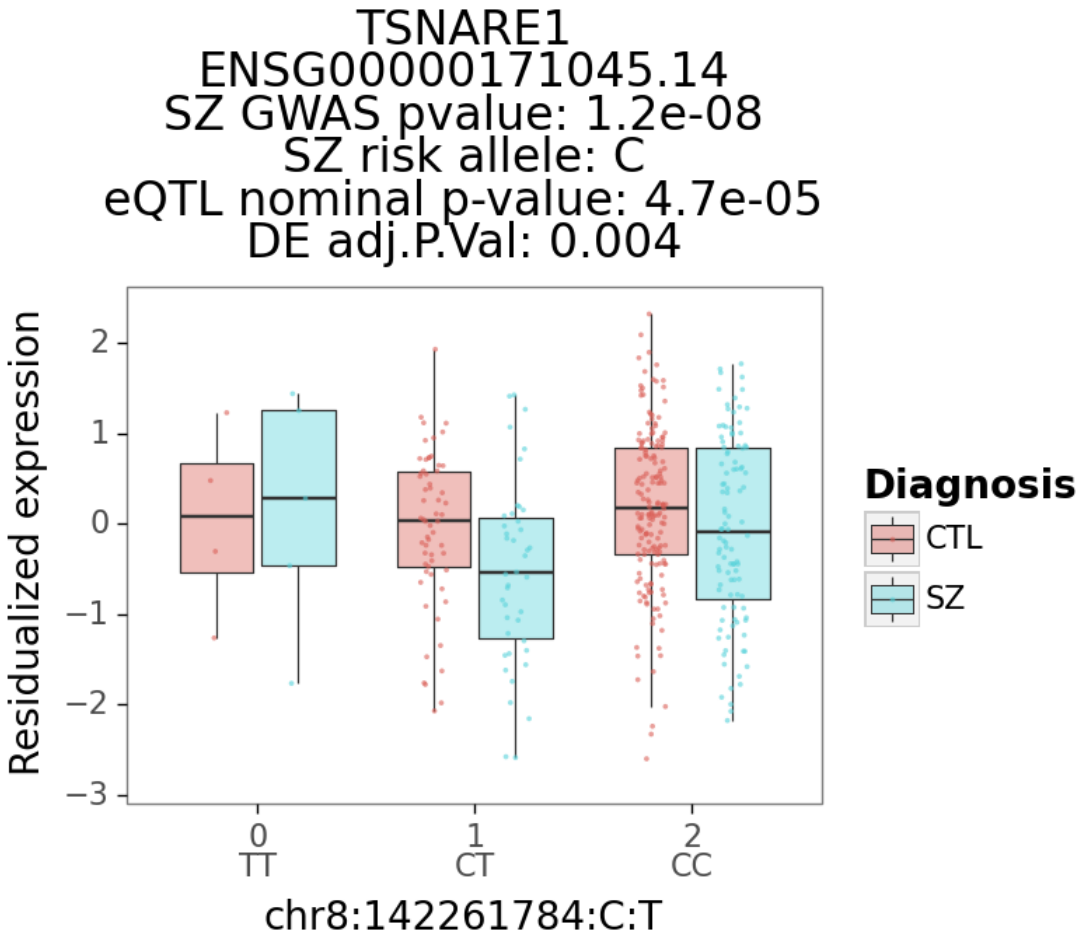

```
<ggplot: (8779624369770)>
```



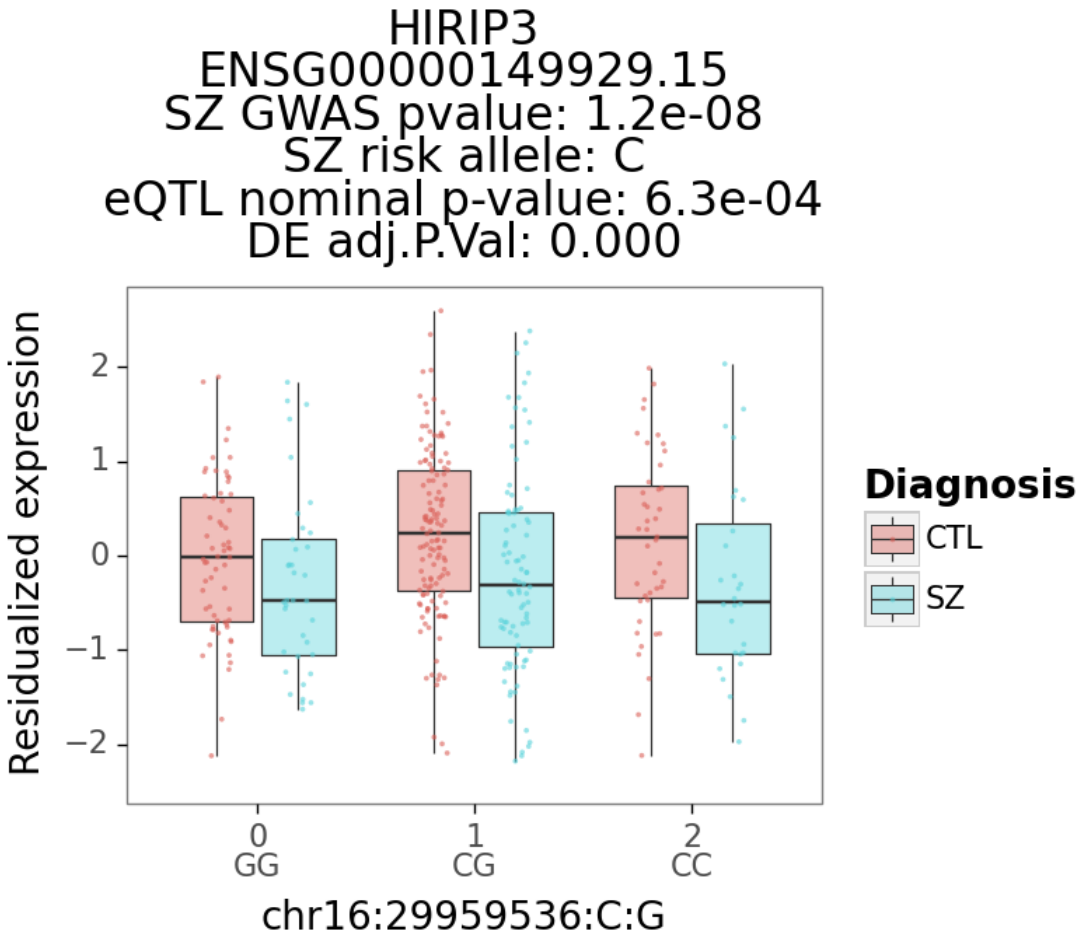

```
<ggplot: (8779627186728)>
```



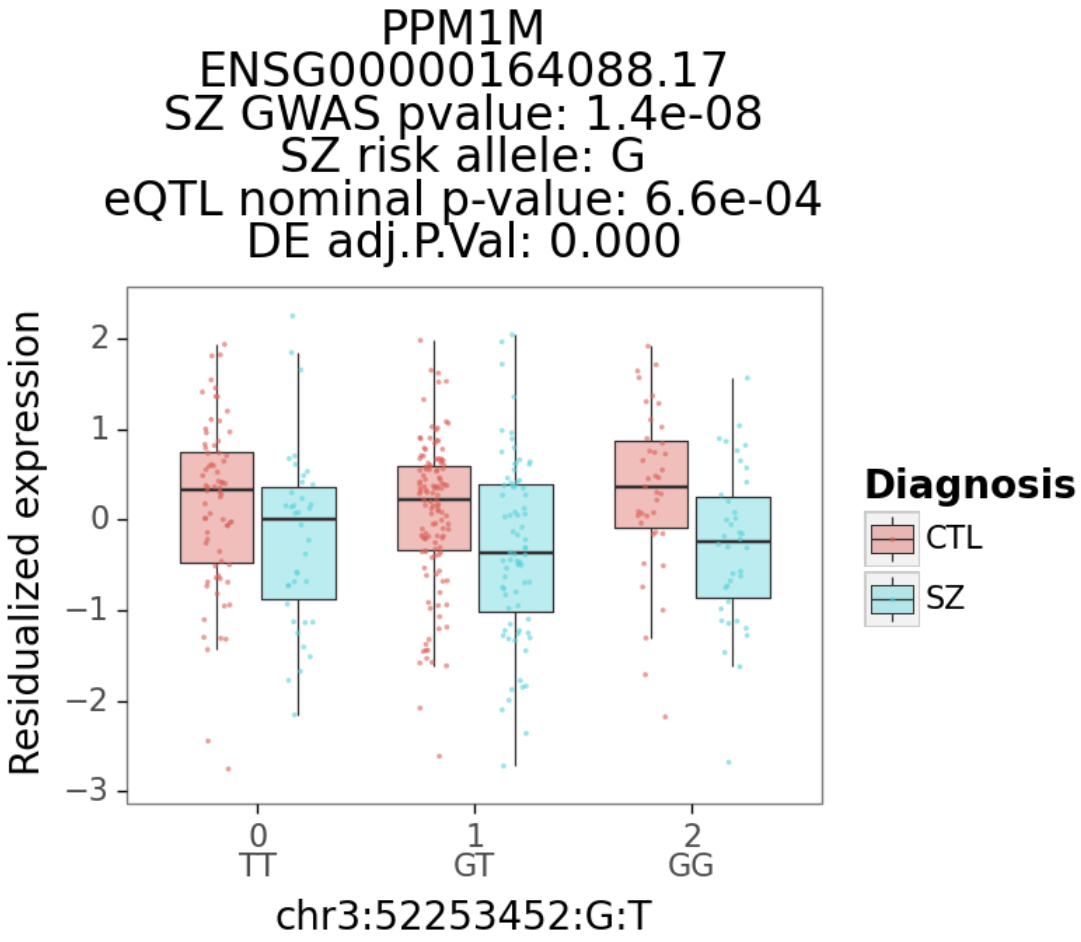

```
<ggplot: (8779617139191)>
```



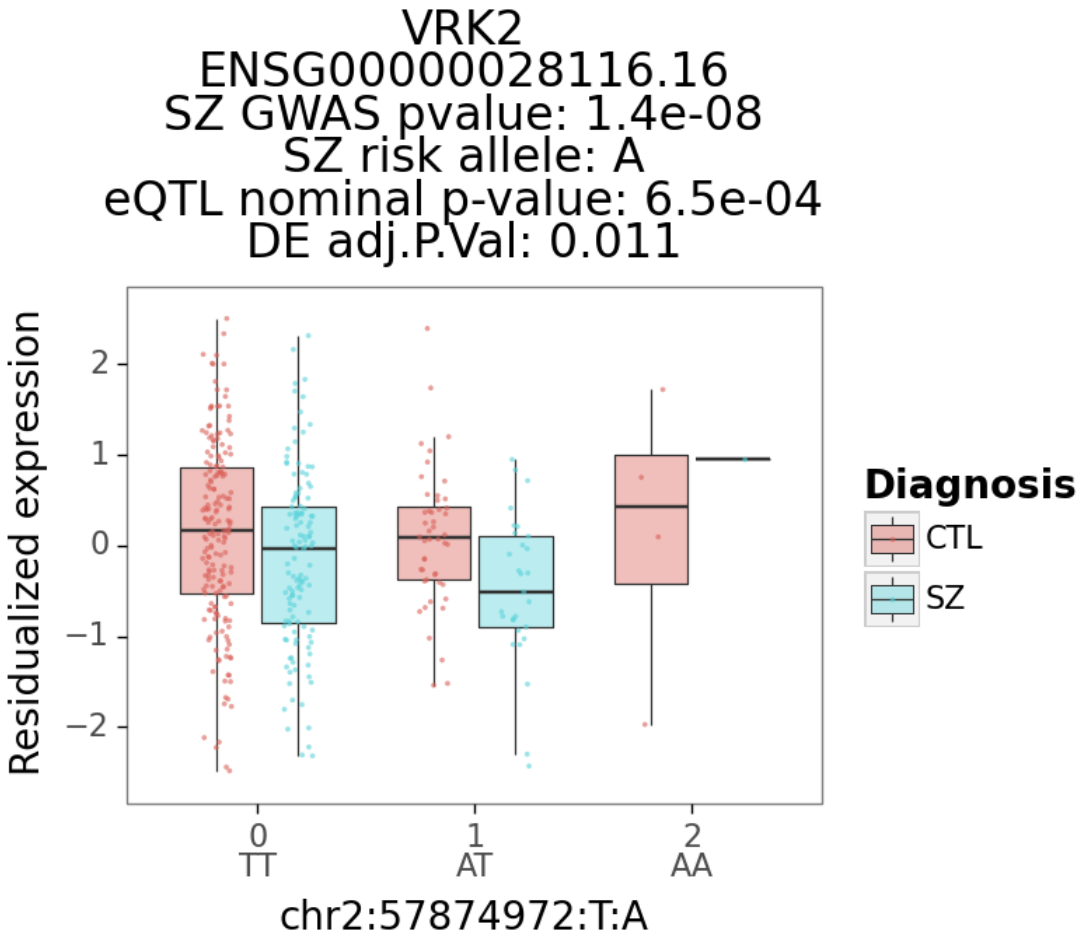

```
<ggplot: (8779616807331)>
```



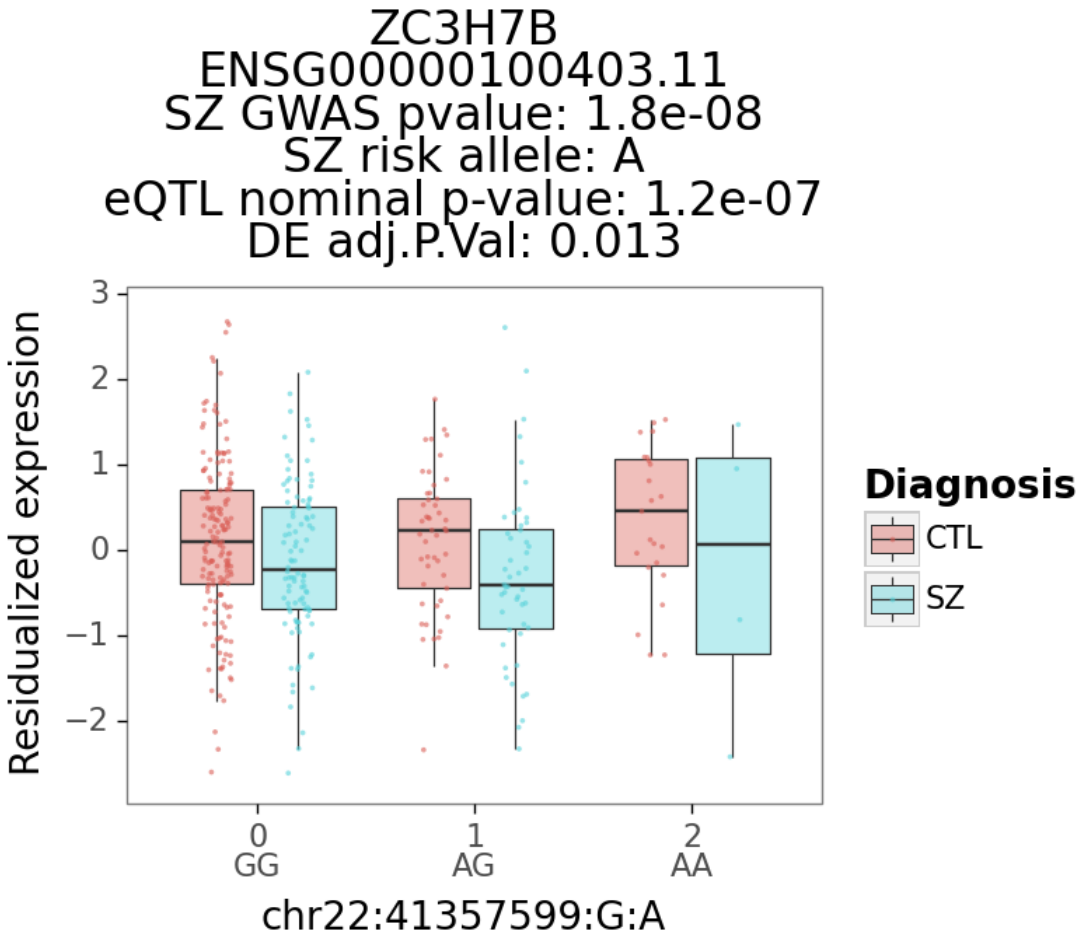

```
<ggplot: (8779648335812)>
```



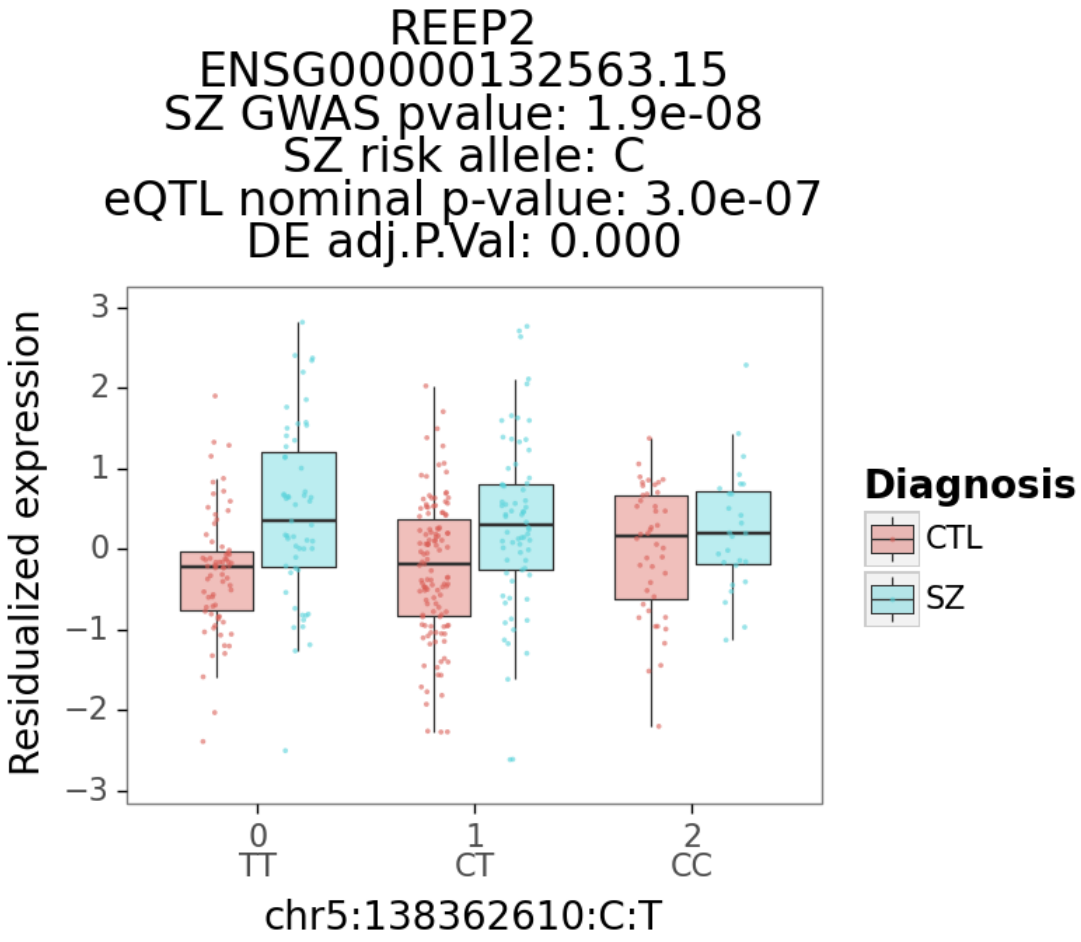
<ggplot: (8779672707127)>



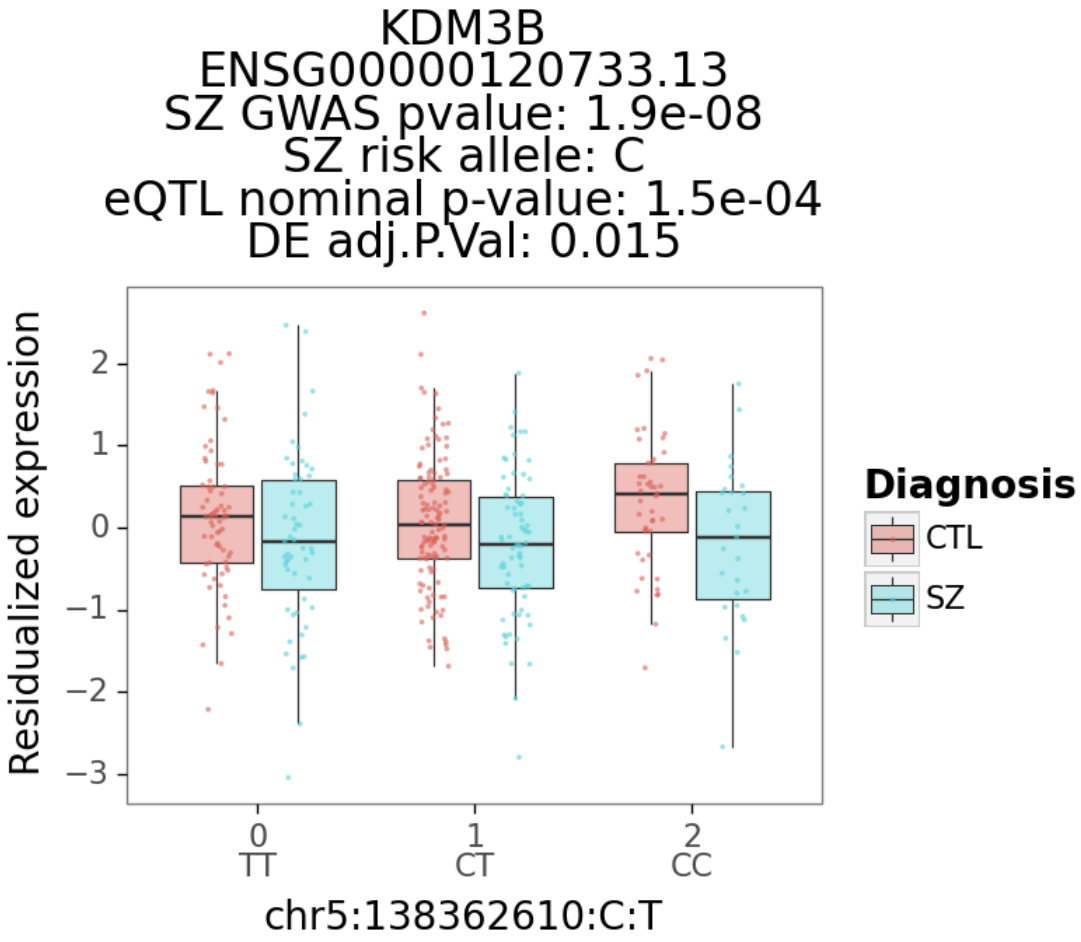

```
<ggplot: (8779674024744)>
```



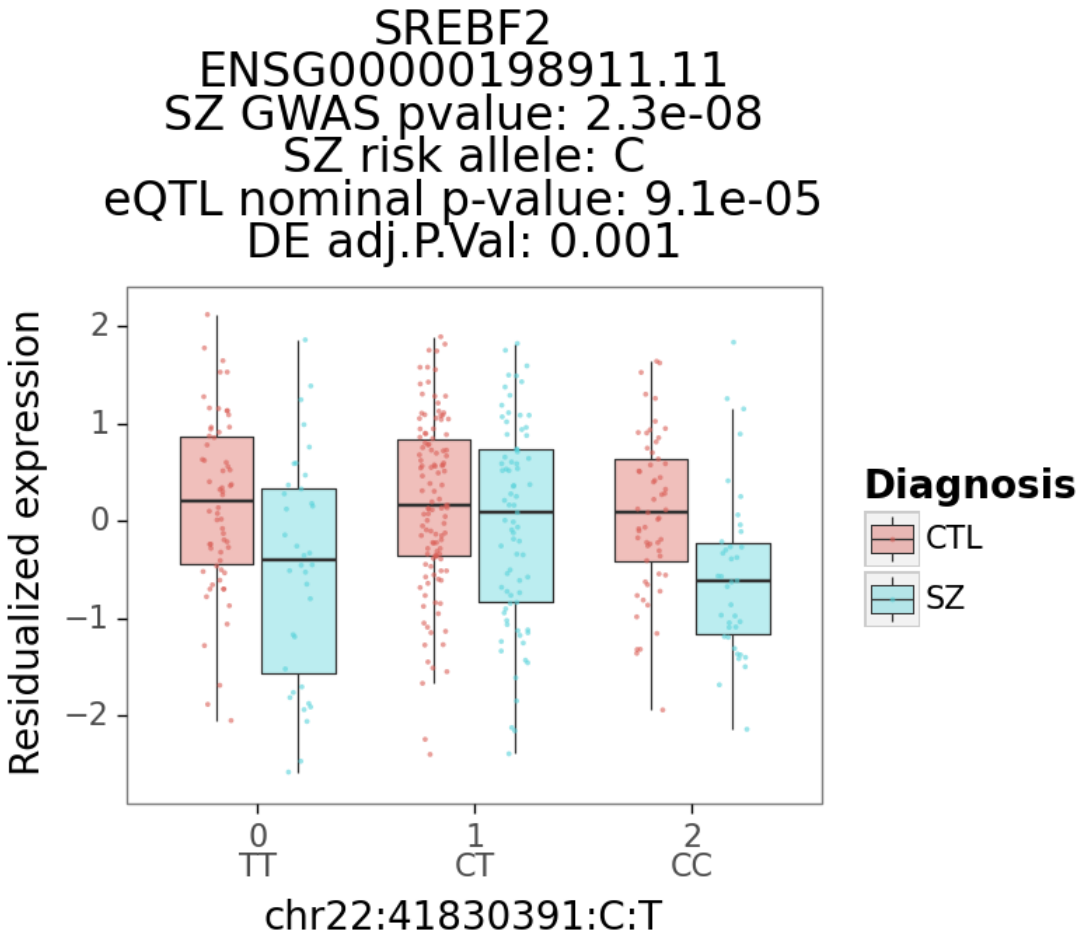

```
<ggplot: (8779673346380)>
```



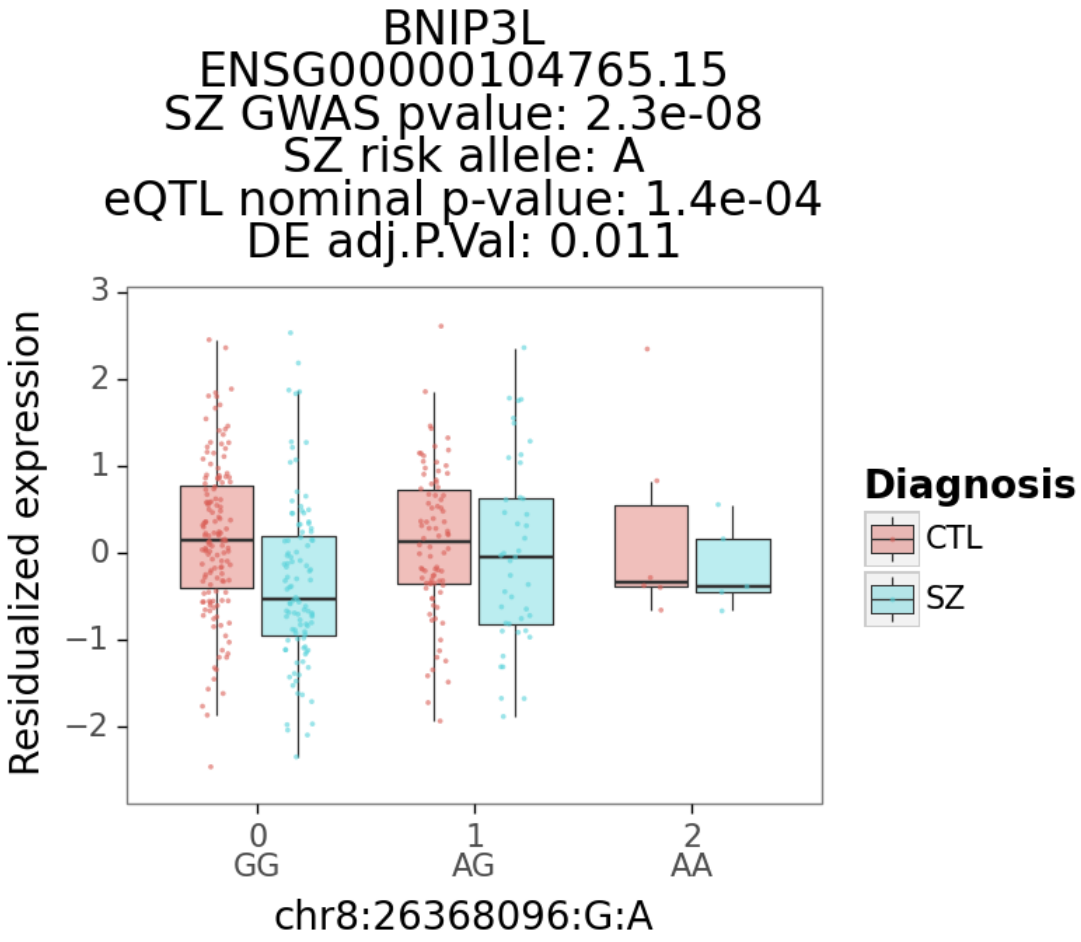

```
<ggplot: (8779649073511)>
```



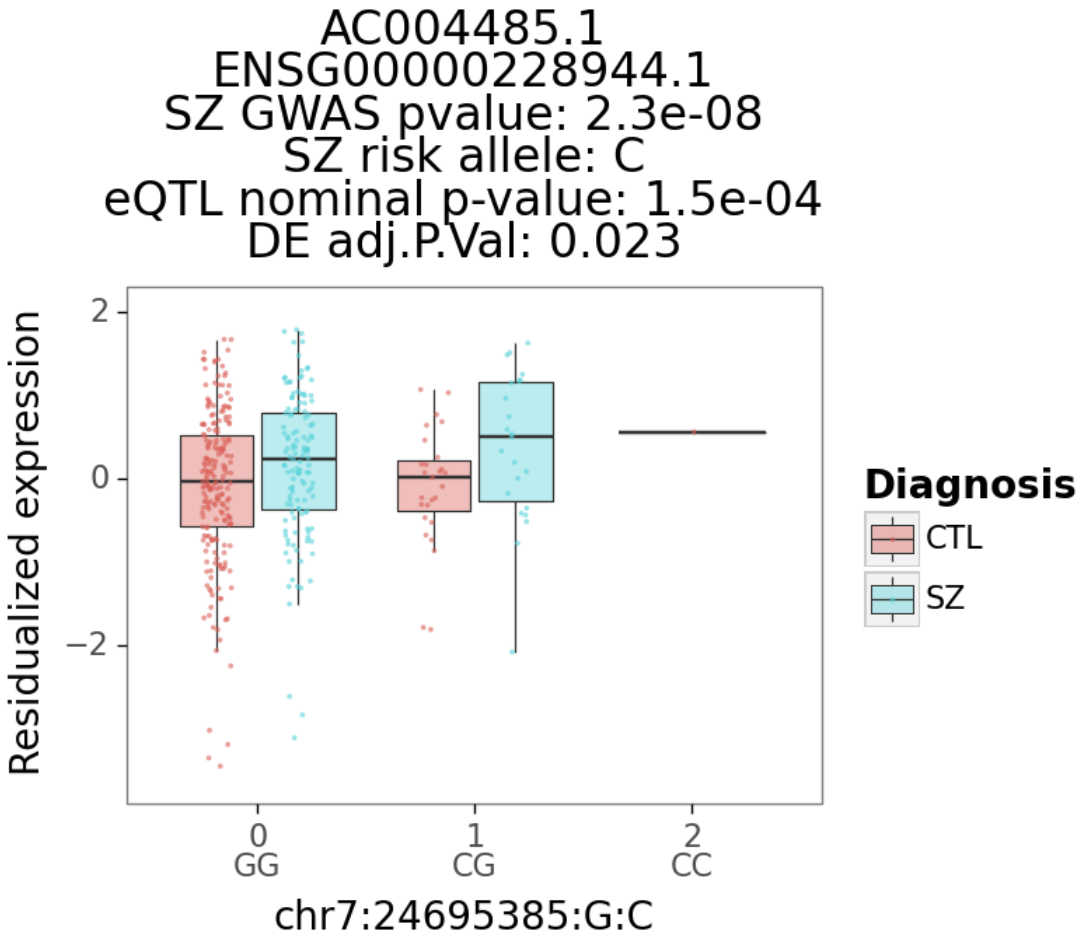

```
<ggplot: (8779647828902)>
```



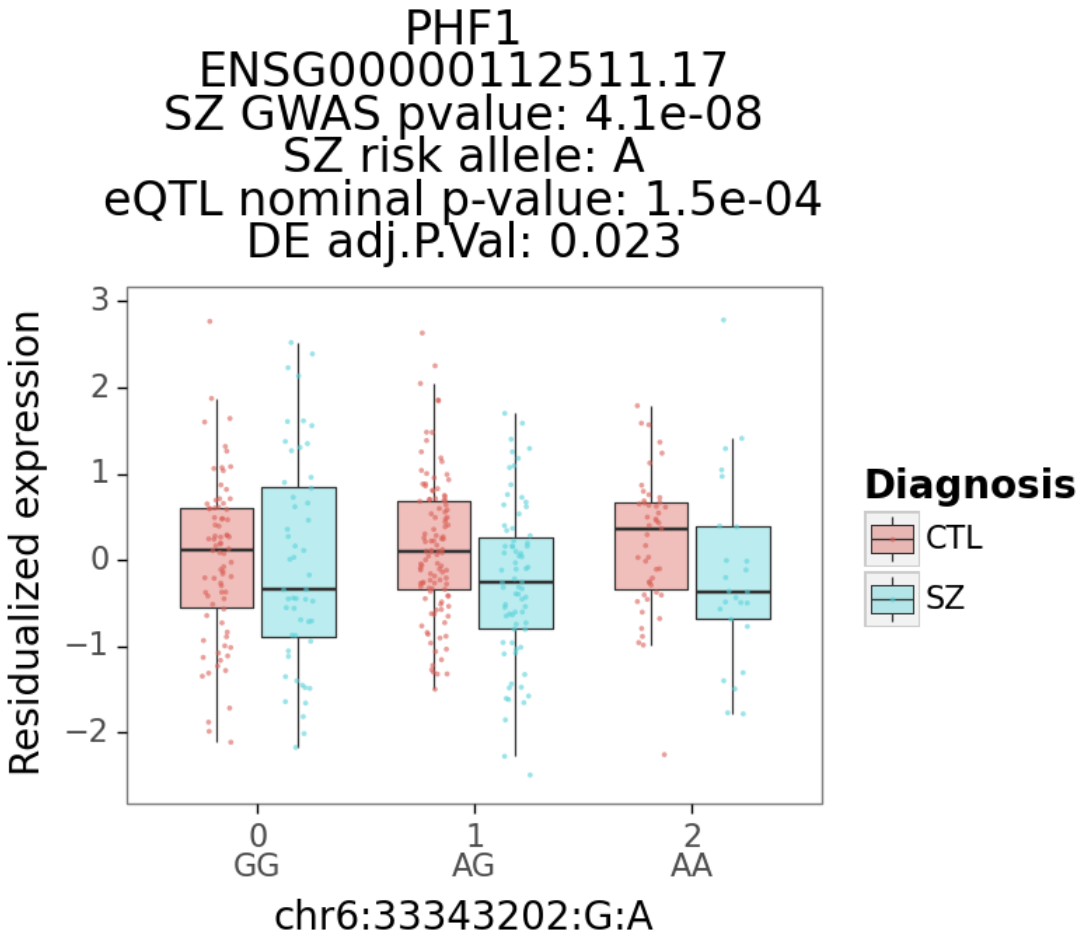

```
<ggplot: (8779647337539)>
```



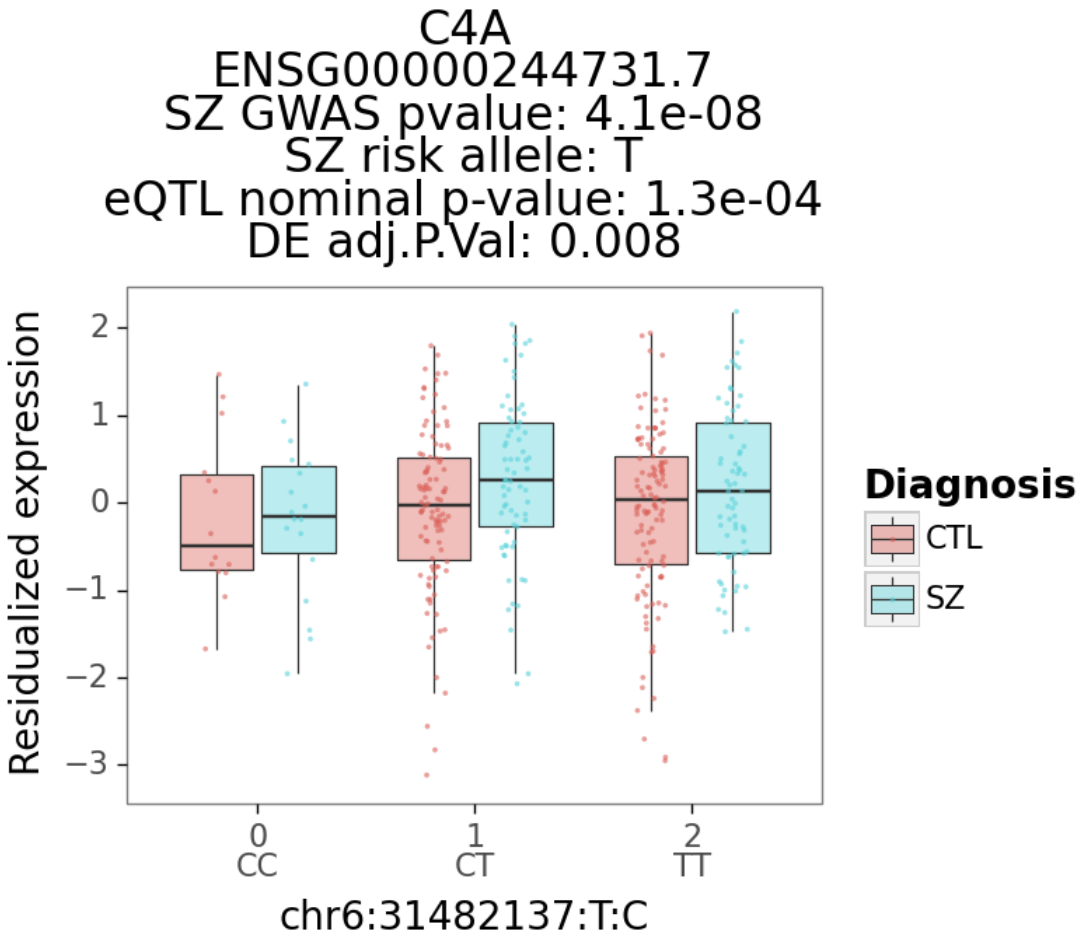

```
<ggplot: (8779672754827)>
```



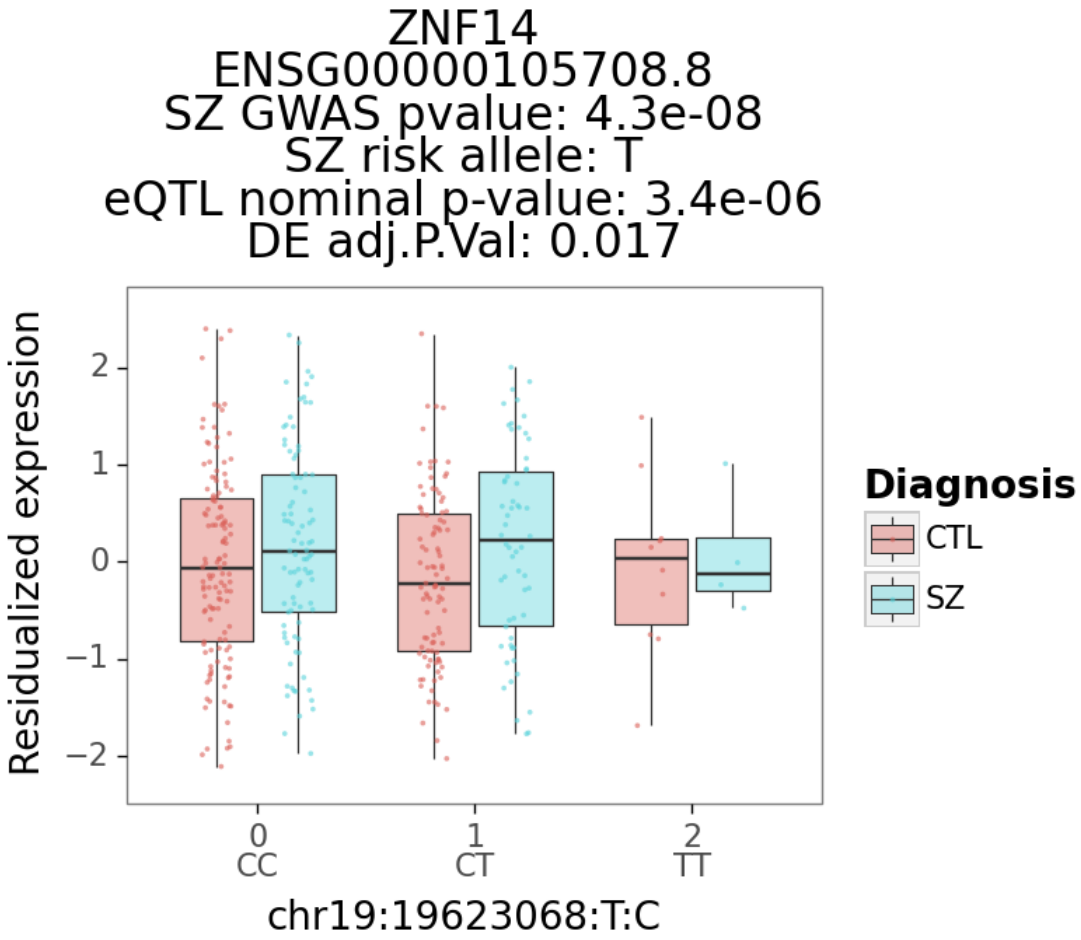

```
<ggplot: (8779673699669)>
```



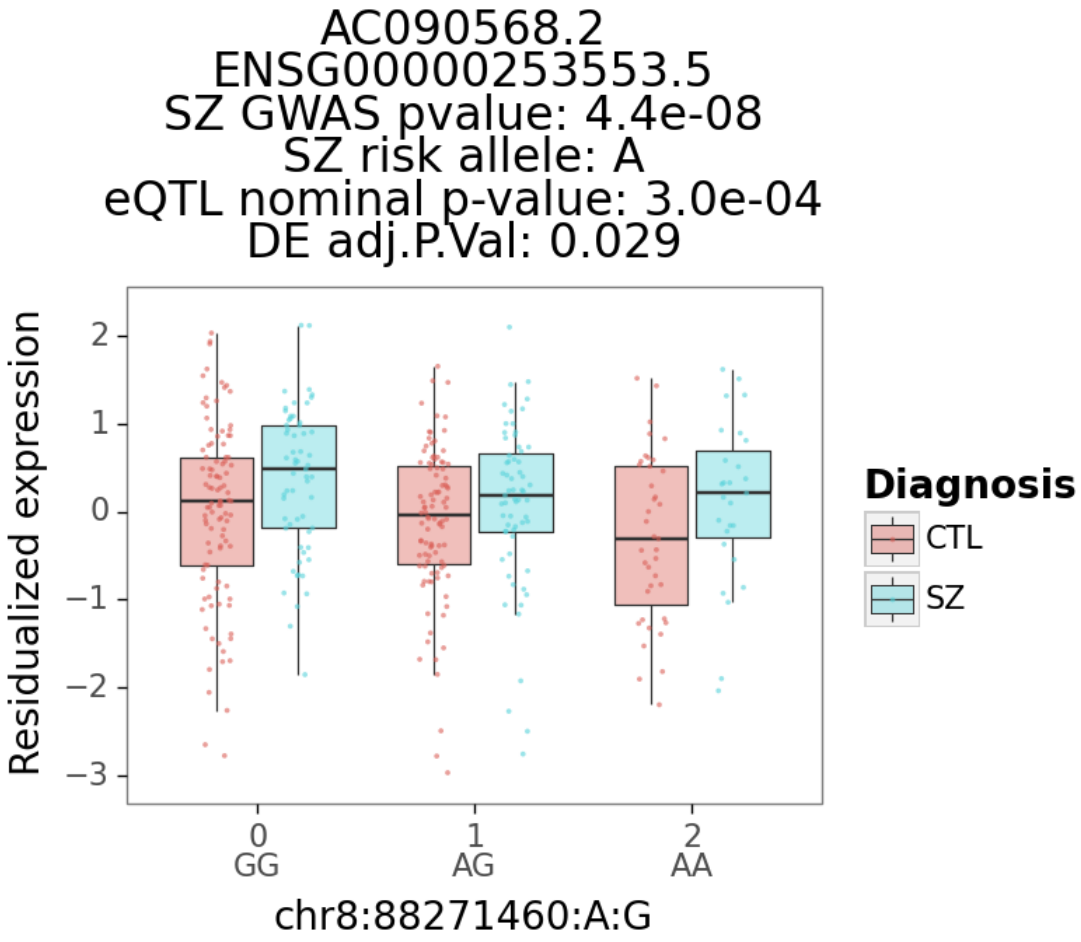

```
<ggplot: (8779673791501)>
```




```
<ggplot: (8779674284858)>
```




```
<ggplot: (8779616381639)>
```




```
<ggplot: (8779673353409)>
```