main

July 11, 2021

1 Tissue comparison for differential expression analysis

[1]:
```
import functools
import numpy as np
import pandas as pd
from plotnine import *
from scipy.stats import binom_test, fisher_exact, linregress

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.*')
```

[2]:
```
config = {
    'caudate': '../../../caudate/_m/genes/diffExpr_maleVfemale_full.txt',
    'dlpfc': '../../../dlpfc/_m/genes/diffExpr_maleVfemale_full.txt',
    'hippo': '../../../hippocampus/_m/genes/diffExpr_maleVfemale_full.txt',
    'cmc_dlpfc': '../../../cmc_dlpfc/_m/genes/diffExpr_maleVfemale_full.txt',
}
```

[3]:
```
@functools.lru_cache()
def get_deg(filename):
    dft = pd.read_csv(filename, sep='\t', index_col=0)
    dft['Feature'] = dft.index
    dft['Dir'] = np.sign(dft['t'])
    if 'gene_id' in dft.columns:
        dft['ensemblID'] = dft.gene_id.str.replace('\\..*', '', regex=True)
    elif 'ensembl_gene_id' in dft.columns:
        dft.rename(columns={'ensembl_gene_id': 'ensemblID'}, inplace=True)
    return dft[['Feature', 'ensemblID', 'adj.P.Val', 'logFC', 't', 'Dir']]


@functools.lru_cache()
def get_deg_sig(filename, fdr):
    dft = get_deg(filename)
    return dft[(dft['adj.P.Val'] < fdr)]
```

```
@functools.lru_cache()
def merge_dataframes(tissue1, tissue2):
    return get_deg(config[tissue1]).merge(get_deg(config[tissue2]),
                                          on='Feature',
                                          suffixes=['_%s' % tissue1, '_%s' %␣
 ↪tissue2])


@functools.lru_cache()
def merge_dataframes_sig(tissue1, tissue2):
    fdr = 0.05
    return get_deg_sig(config[tissue1], fdr).merge(get_deg_sig(config[tissue2],␣
 ↪fdr),
                                                   on='Feature',
                                                   suffixes=['_%s' % tissue1,␣
 ↪'_%s' % tissue2])
```

[4]:
```
def enrichment_binom(tissue1, tissue2, merge_fnc):
    df = merge_fnc(tissue1, tissue2)
    df['agree'] = df['Dir_%s' % tissue1] * df['Dir_%s' % tissue2]
    dft = df.groupby('agree').size().reset_index()
    print(dft)
    return binom_test(dft[0].iloc[1], dft[0].sum()) if dft.shape[0] != 1 else␣
 ↪print("All directions agree!")


def cal_fishers(tissue1, tissue2):
    df = merge_dataframes(tissue1, tissue2)
    fdr = 0.05
    table = [[np.sum((df['adj.P.Val_%s' % tissue1]<fdr) &
                     ((df['adj.P.Val_%s' % tissue2]<fdr))),
              np.sum((df['adj.P.Val_%s' % tissue1]<fdr) &
                     ((df['adj.P.Val_%s' % tissue2]>=fdr)))],
             [np.sum((df['adj.P.Val_%s' % tissue1]>=fdr) &
                     ((df['adj.P.Val_%s' % tissue2]<fdr))),
              np.sum((df['adj.P.Val_%s' % tissue1]>=fdr) &
                     ((df['adj.P.Val_%s' % tissue2]>=fdr)))]]
    print(table)
    return fisher_exact(table)


def calculate_corr(xx, yy):
    '''This calculates R^2 correlation via linear regression:
         - used to calculate relationship between 2 arrays
         - the arrays are principal components 1 or 2 (PC1, PC2) AND gender
```

```
        - calculated on a scale of 0 to 1 (with 0 being no correlation)
      Inputs:
        x: array of Gender (converted to binary output)
        y: array of PC
      Outputs:
        1. r2
        2. p-value, two-sided test
          - whose null hypothesis is that two sets of data are uncorrelated
        3. slope (beta): directory of correlations
    '''
    slope, intercept, r_value, p_value, std_err = linregress(xx, yy)
    return r_value, p_value


def corr_annotation(tissue1, tissue2, merge_fnc):
    dft = merge_fnc(tissue1, tissue2)
    xx = dft['t_%s' % tissue1]
    yy = dft['t_%s' % tissue2]
    r_value1, p_value1 = calculate_corr(xx, yy)
    return 'R2: %.2f\nP-value: %.2e' % (r_value1**2, p_value1)


def tissue_annotation(tissue):
    return {'dlpfc': 'DLPFC', 'hippo': 'Hippocampus',
            'caudate': 'Caudate', 'cmc_dlpfc': 'CMC DLPFC'}[tissue]
```

[5]:
```
def plot_corr_impl(tissue1, tissue2, merge_fnc):
    dft = merge_fnc(tissue1, tissue2)
    title = '\n'.join([corr_annotation(tissue1, tissue2, merge_fnc)])
    xlab = 'T-statistic (%s)' % tissue_annotation(tissue1)
    ylab = 'T-statistic (%s)' % tissue_annotation(tissue2)
    pp = ggplot(dft, aes(x='t_%s'%tissue1, y='t_%s' % tissue2))\
    + geom_point(alpha=0.75, size=3)\
    + theme_matplotlib()\
    + theme(axis_text=element_text(size=18),
            axis_title=element_text(size=20, face='bold'),
            plot_title=element_text(size=22))
    pp += labs(x=xlab, y=ylab, title=title)
    return pp


def plot_corr(tissue1, tissue2, merge_fnc):
    return plot_corr_impl(tissue1, tissue2, merge_fnc)


def save_plot(p, fn, width=7, height=7):
    '''Save plot as svg, png, and pdf with specific label and dimension.'''
```

```
    for ext in ['.svg', '.png', '.pdf']:
        p.save(fn+ext, width=width, height=height)
```

1.1 Sample summary

[6]:
```
pheno_file = '/ceph/projects/v3_phase3_paper/inputs/phenotypes/merged/_m/
 ↪merged_phenotypes.csv'
pheno = pd.read_csv(pheno_file, index_col=0)
pheno = pheno[(pheno['Age'] > 17) &
              (pheno['Dx'].isin(['Schizo', 'Control'])) &
              (pheno["Race"].isin(["AA", "CAUC"]))].copy()
pheno.head(2)
```

[6]:

	BrNum	RNum	Region	RIN	Age	Sex	Race	Dx
R12864	Br1303	R12864	Caudate	9.6	42.98	F	AA	Schizo
R12865	Br1320	R12865	Caudate	9.5	53.12	M	AA	Schizo

[7]:
```
pheno.groupby(['Region']).size()
```

[7]:
```
Region
Caudate    394
DLPFC      360
HIPPO      376
dtype: int64
```

[8]:
```
pheno.groupby(['Region', 'Sex']).size()
```

[8]:
```
Region   Sex
Caudate  F      121
         M      273
DLPFC    F      114
         M      246
HIPPO    F      121
         M      255
dtype: int64
```

1.2 BrainSeq Tissue Comparison

[9]:
```
caudate = get_deg(config['caudate'])
caudate.groupby('Dir').size()
```

[9]:
```
Dir
-1.0    11133
 1.0    12355
dtype: int64
```

[10]:
```
caudate[(caudate['adj.P.Val'] < 0.05)].shape
```

[10]: (380, 6)

[11]:
```
dlpfc = get_deg(config['dlpfc'])
dlpfc.groupby('Dir').size()
```

[11]:
```
Dir
-1.0    11240
 1.0    11799
dtype: int64
```

[12]:
```
dlpfc[(dlpfc['adj.P.Val'] < 0.05)].shape
```

[12]: (573, 6)

[13]:
```
hippo = get_deg(config['hippo'])
hippo.groupby('Dir').size()
```

[13]:
```
Dir
-1.0    11840
 1.0    11150
dtype: int64
```

[14]:
```
hippo[(hippo['adj.P.Val'] < 0.05)].shape
```

[14]: (105, 6)

1.2.1 Enrichment of DEG

[15]:
```
cal_fishers('caudate', 'dlpfc')
```

```
[[117, 236], [428, 21171]]
```

[15]: (24.52287937589102, 3.2462962516016504e-100)

[16]:
```
cal_fishers('caudate', 'hippo')
```

```
[[85, 270], [16, 21688]]
```

[16]: (426.73148148148147, 9.812269867491168e-140)

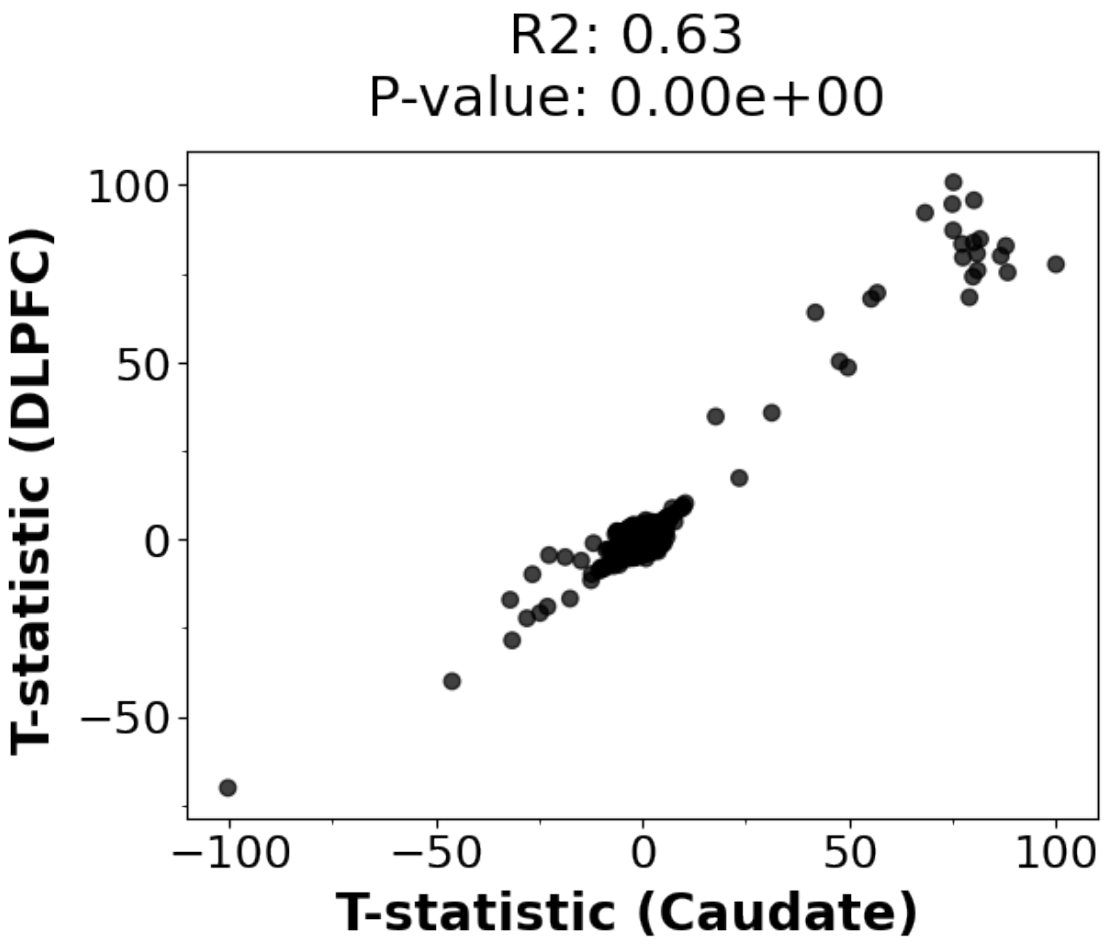
[17]:
```
cal_fishers('dlpfc', 'hippo')
```

```
[[81, 474], [18, 21662]]
```

[17]: (205.65189873417722, 6.409439839482686e-114)

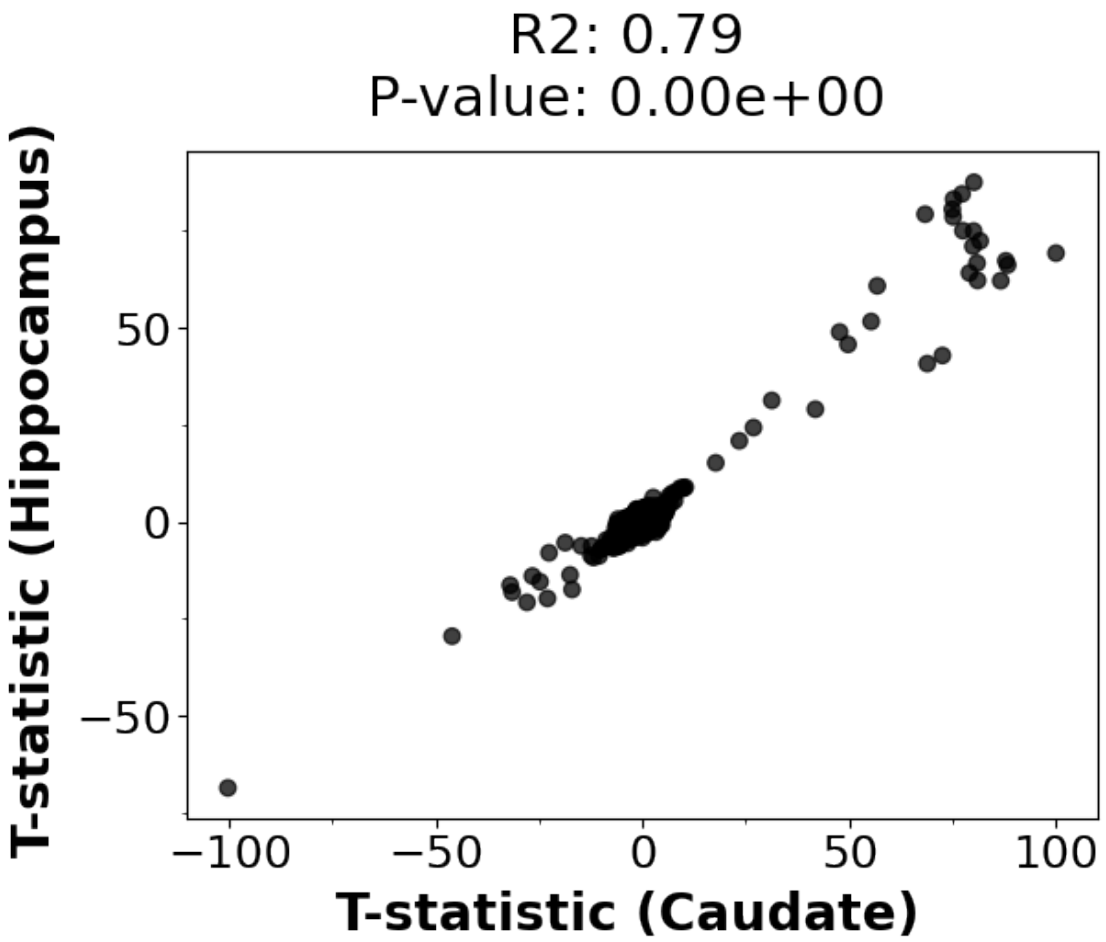
1.2.2 Correlation

[18]:
```
pp = plot_corr('caudate', 'dlpfc', merge_dataframes)
pp
```



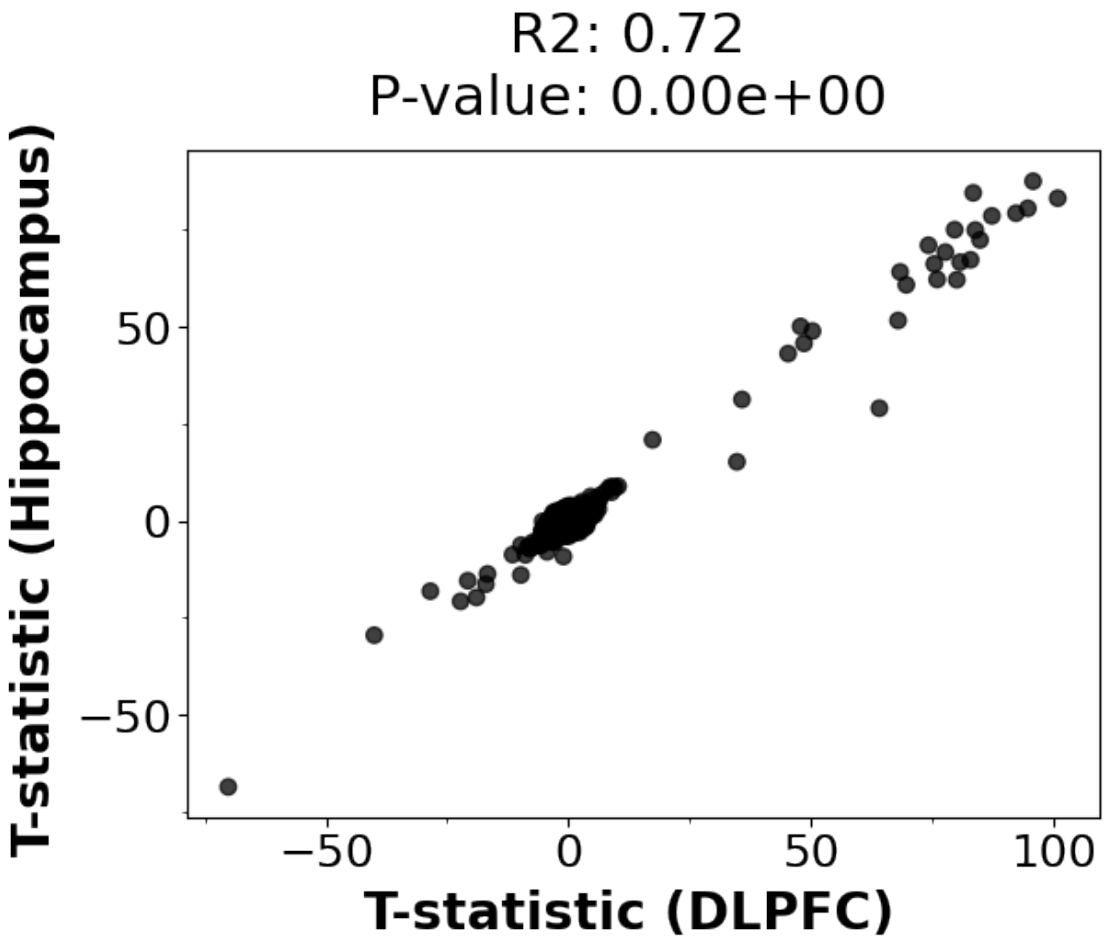
[18]:
```
<ggplot: (8758747217731)>
```

[19]:
```
qq = plot_corr('caudate', 'hippo', merge_dataframes)
qq
```



[19]: <ggplot: (8757995576664)>

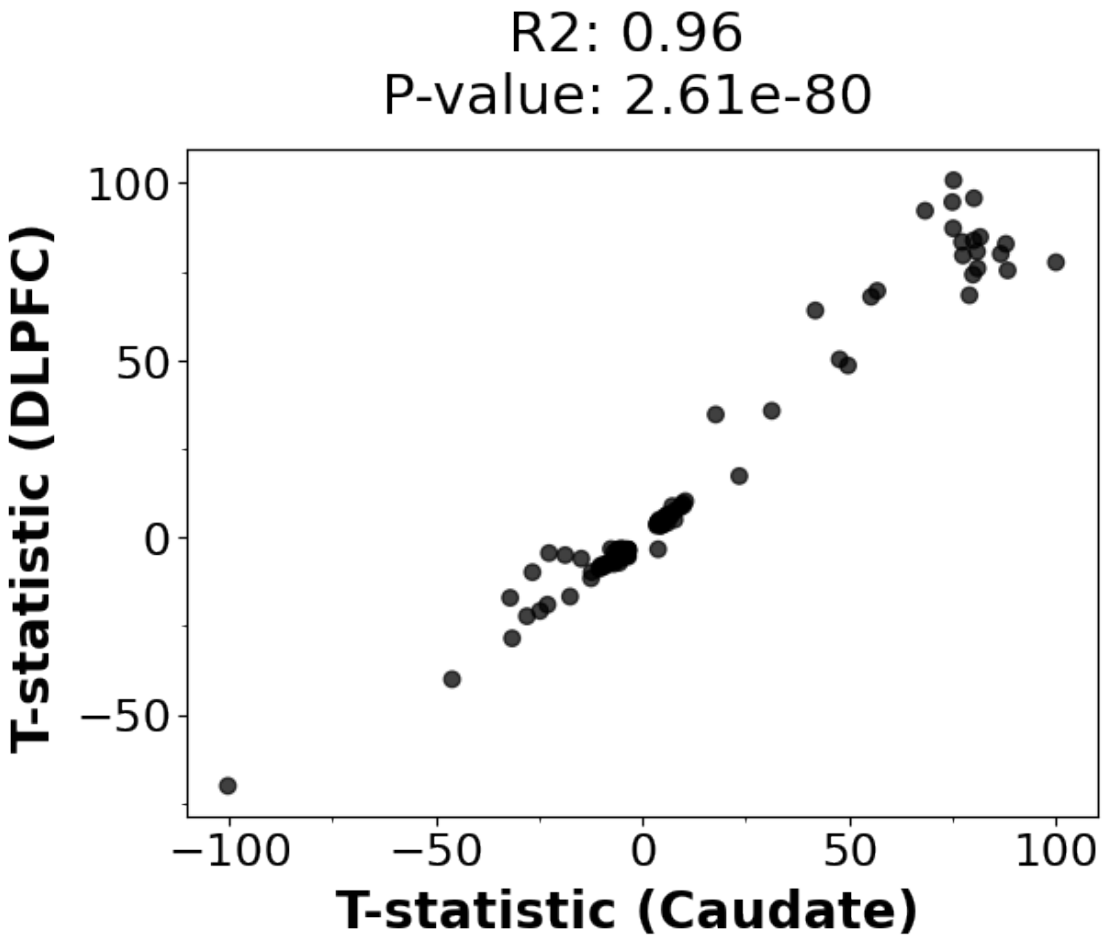
[20]:
```
ww = plot_corr('dlpfc', 'hippo', merge_dataframes)
ww
```



[20]: <ggplot: (8757995634920)>

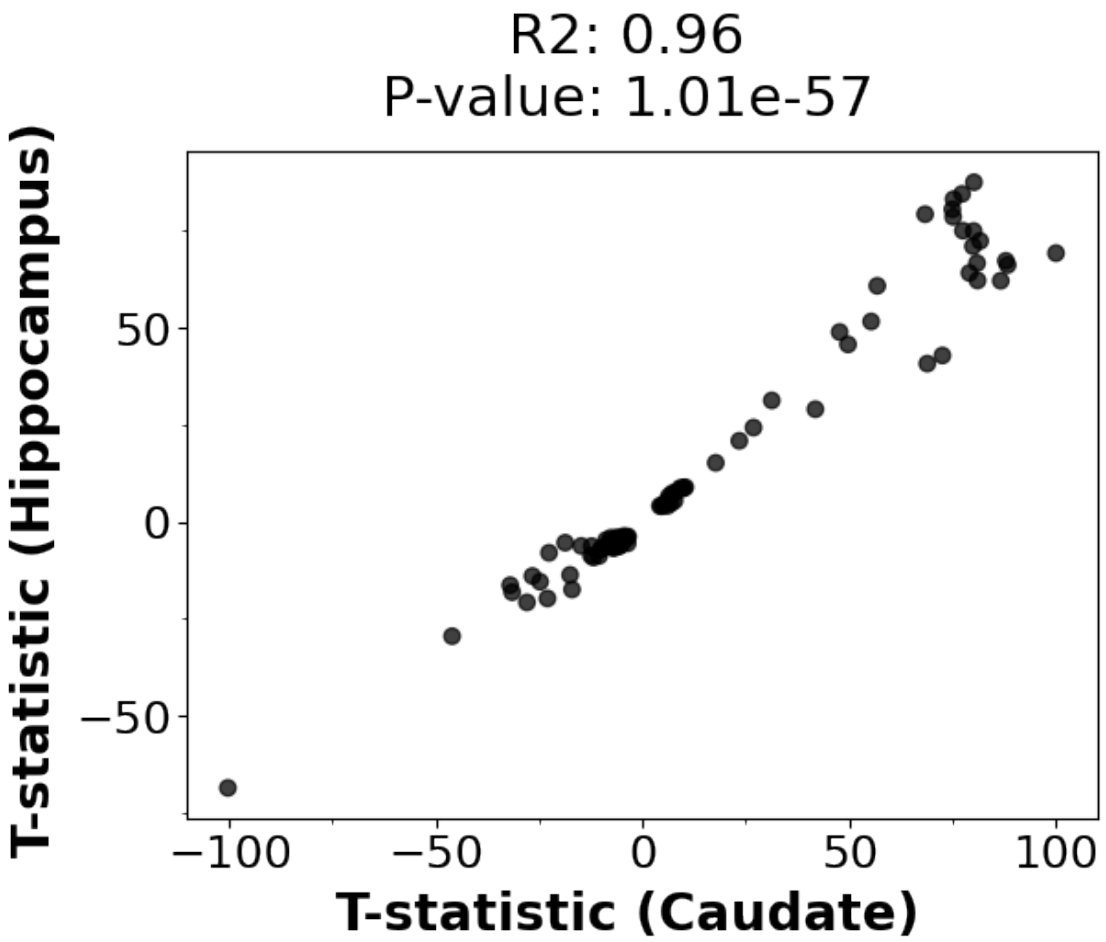
1.2.3 Significant correlation, FDR < 0.05

[21]:
```
pp = plot_corr('caudate', 'dlpfc', merge_dataframes_sig)
pp
```



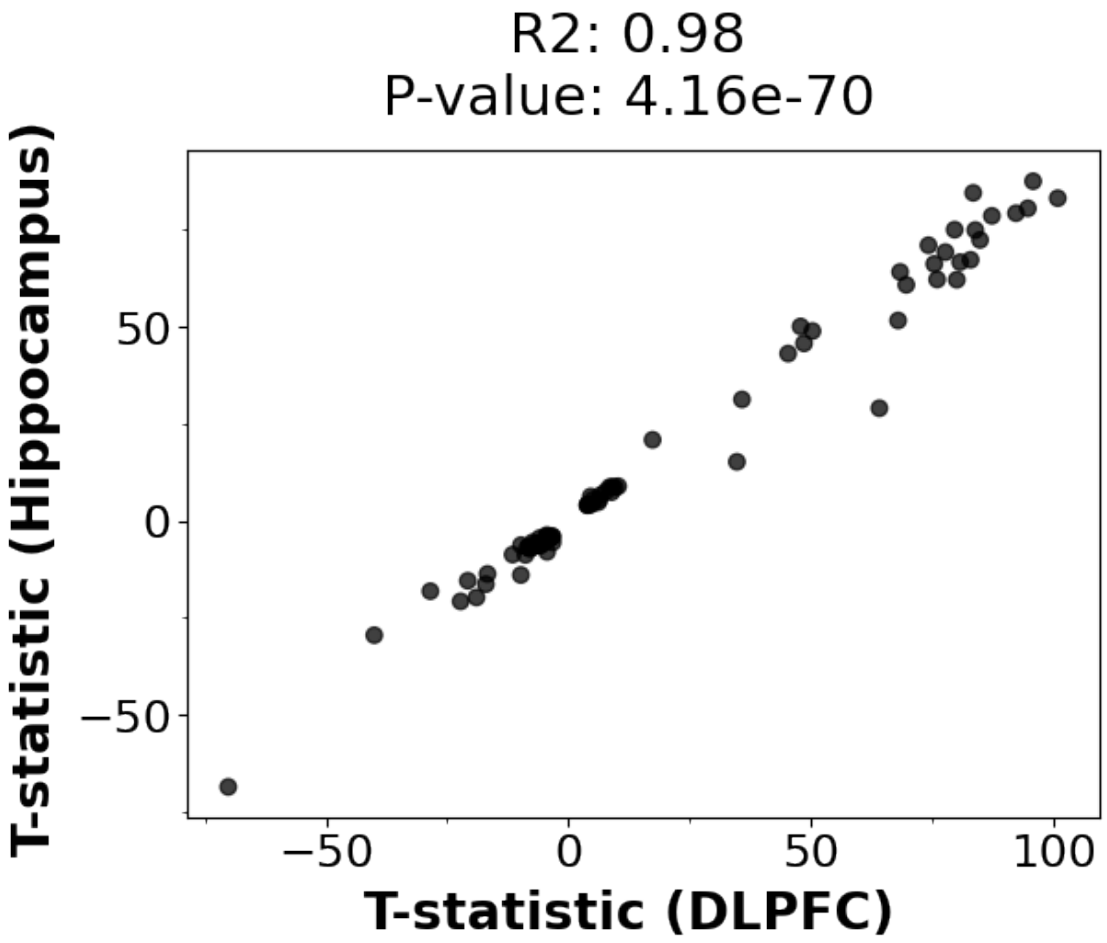
[21]:
```
<ggplot: (8757995338560)>
```

[22]:
```
qq = plot_corr('caudate', 'hippo', merge_dataframes_sig)
qq
```



[22]: <ggplot: (8757995338725)>

[23]:
```
ww = plot_corr('dlpfc', 'hippo', merge_dataframes_sig)
ww
```



[23]:
```
<ggplot: (8757995338716)>
```

[24]:
```
#save_plot(pp, 'dlpfc_caudate_tstatistic_corr_sig')
#save_plot(qq, 'hippo_caudate_tstatistic_corr_sig')
#save_plot(ww, 'hippo_dlpfc_tstatistic_corr_sig')
```

1.2.4 Directionality test

All genes

[25]:
```
enrichment_binom('caudate', 'dlpfc', merge_dataframes)
```

```
   agree      0
0   -1.0   9764
1    1.0  12188
```

[25]:
```
3.115367597709529e-60
```

[26]:
```
enrichment_binom('caudate', 'hippo', merge_dataframes)
```

```
   agree      0
0   -1.0   7835
1    1.0  14224
```

[26]:
```
5e-324
```

[27]:
```
enrichment_binom('dlpfc', 'hippo', merge_dataframes)
```

```
   agree      0
0   -1.0   8879
1    1.0  13356
```

[27]:
```
2.6476758684712667e-199
```

Significant DEG (FDR < 0.05)

[28]:
```
enrichment_binom('caudate', 'dlpfc', merge_dataframes_sig)
```

```
   agree    0
0   -1.0    1
1    1.0  116
```

[28]:
```
1.420373333985586e-33
```

[29]:
```
df = merge_dataframes_sig("caudate", "dlpfc")
df[(df['agree']<0)]
```

[29]:
```
                Feature ensemblID_caudate  adj.P.Val_caudate  logFC_caudate  \
101  ENSG00000066629.16   ENSG00000066629           0.014861       0.089089

     t_caudate  Dir_caudate  ensemblID_dlpfc  adj.P.Val_dlpfc  logFC_dlpfc  \
101   3.830688          1.0  ENSG00000066629         0.043284    -0.080965

      t_dlpfc  Dir_dlpfc  agree
101 -3.361968       -1.0   -1.0
```

[30]:
```
enrichment_binom('caudate', 'hippo', merge_dataframes_sig)
```

```
   agree   0
0    1.0  85
All directions agree!
```

[31]:
```
enrichment_binom('dlpfc', 'hippo', merge_dataframes_sig)
```

```
   agree   0
0    1.0  81
All directions agree!
```

1.3 Common Mind Comparison

[32]:
```
cmc_dlpfc = get_deg(config['cmc_dlpfc'])
cmc_dlpfc.groupby('Dir').size()
```

[32]:
```
Dir
-1.0    10915
 1.0     9705
dtype: int64
```

[33]:
```
cmc_dlpfc[(cmc_dlpfc['adj.P.Val'] < 0.05)].shape
```

[33]:
```
(1315, 6)
```

1.3.1 Enrichment of DEG

[34]:
```
cal_fishers('dlpfc', 'cmc_dlpfc')
```

```
[[77, 417], [1153, 16427]]
```

[34]:
```
(2.630774478422466, 5.94044306686282e-12)
```

[35]:
```
cal_fishers('hippo', 'cmc_dlpfc')
```

```
[[1, 69], [1225, 16675]]
```

[35]:
```
(0.19727891156462585, 0.09098898206466205)
```

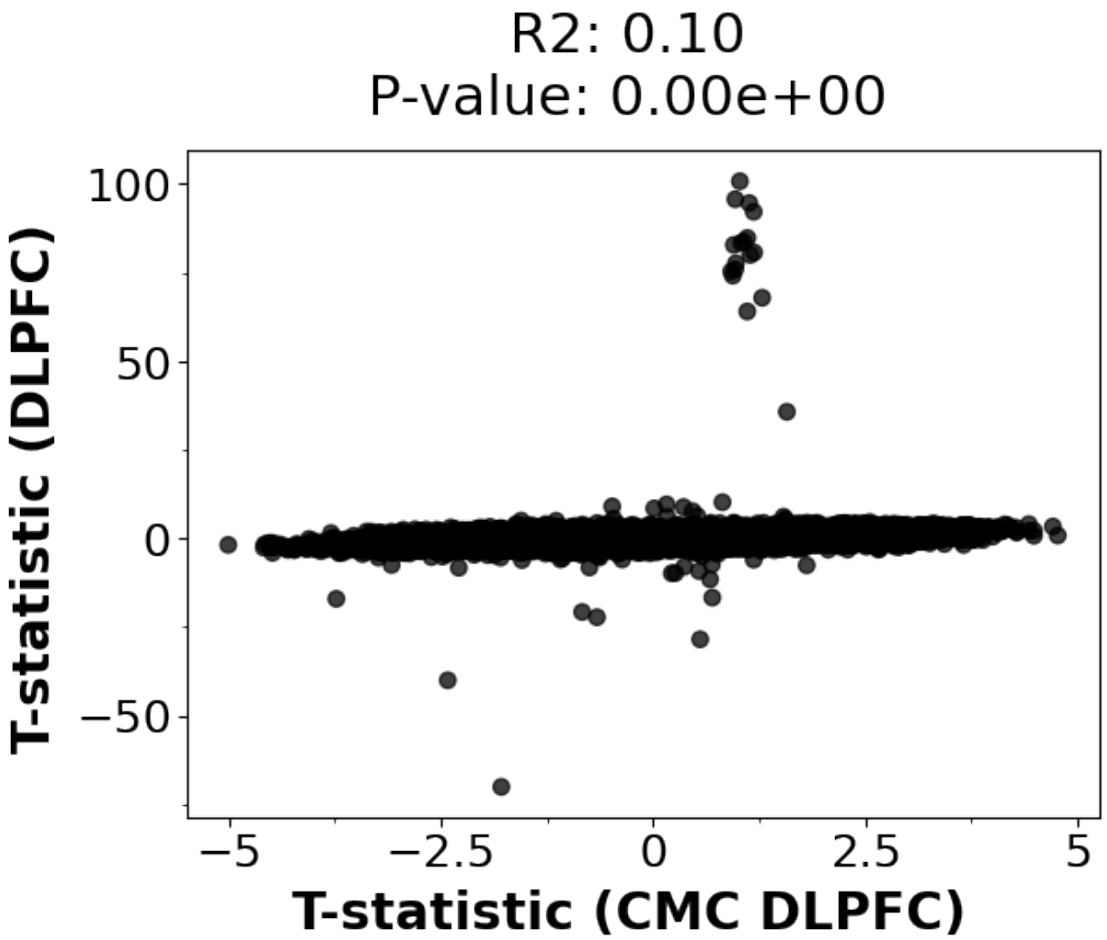
[36]:
```
cal_fishers('caudate', 'cmc_dlpfc')
```

```
[[19, 286], [1210, 16365]]
```

[36]:
```
(0.898500260070508, 0.7326114750021963)
```

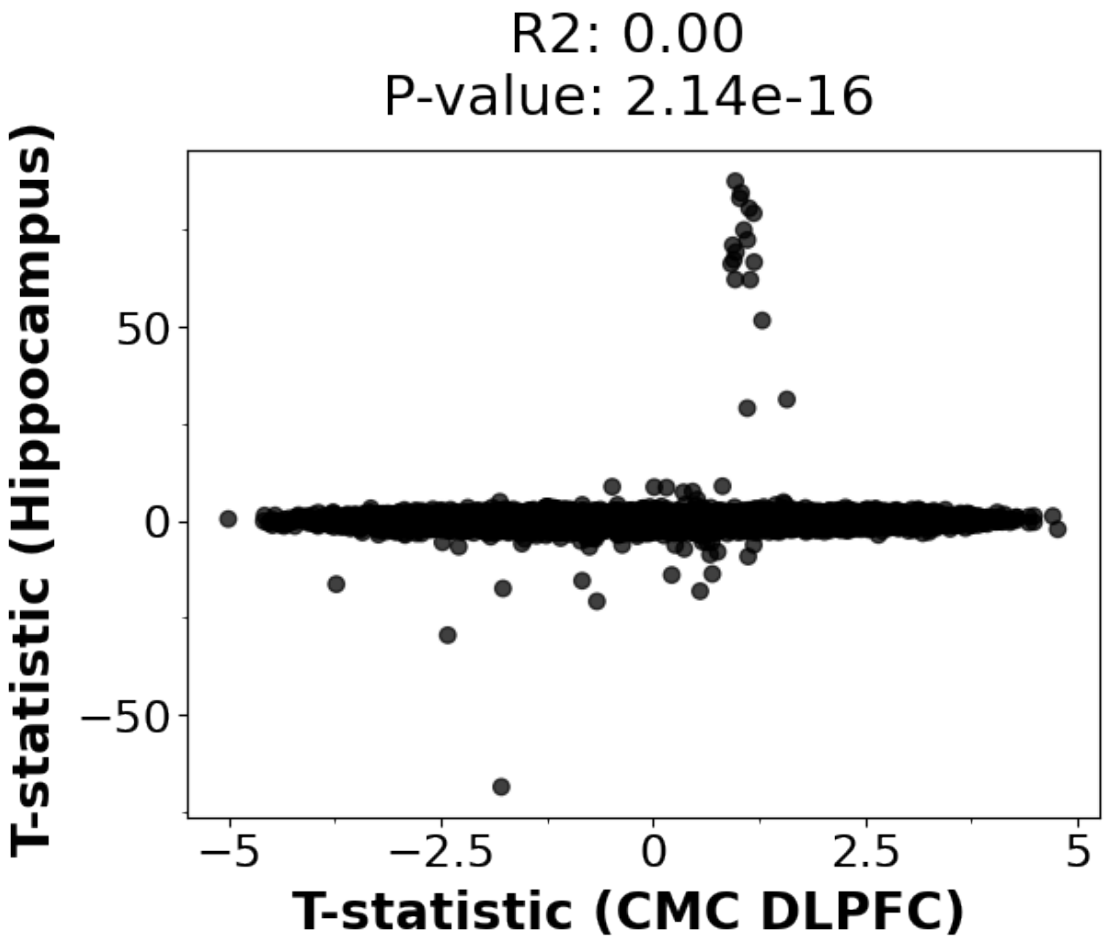
1.3.2 Correlation

[37]:
```
pp = plot_corr('cmc_dlpfc', 'dlpfc', merge_dataframes)
pp
```



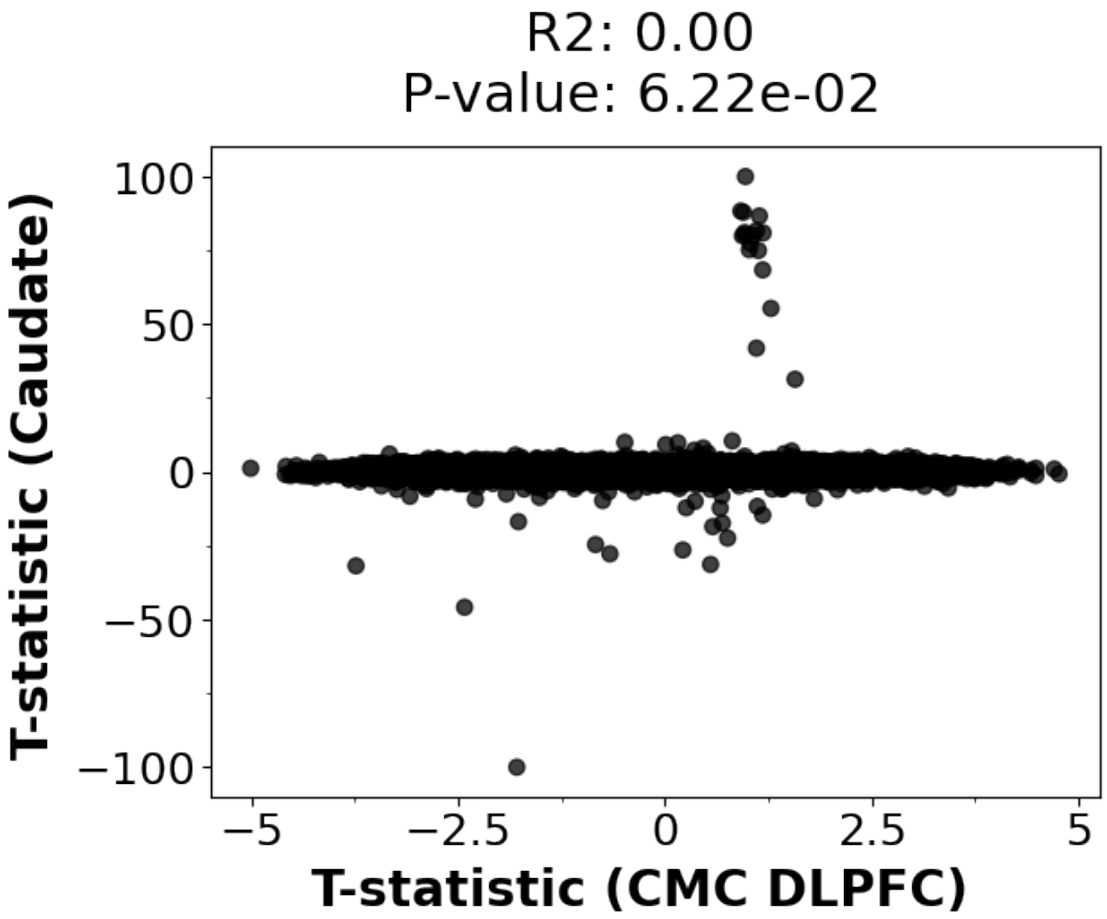
[37]:
```
<ggplot: (8757995337454)>
```

[38]:
```
qq = plot_corr('cmc_dlpfc', 'hippo', merge_dataframes)
qq
```



[38]:
```
<ggplot: (8757994357679)>
```

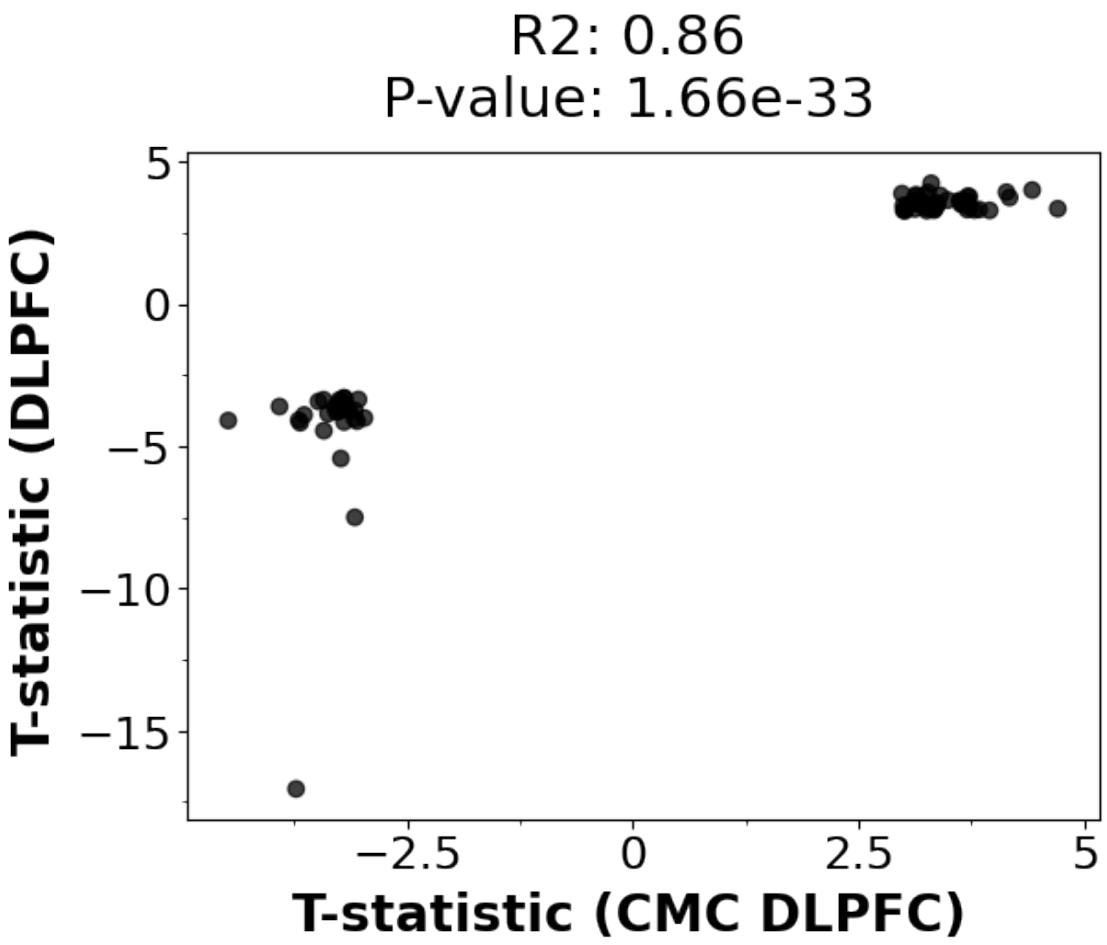
[39]:
```
ww = plot_corr('cmc_dlpfc', 'caudate', merge_dataframes)
ww
```



[39]: <ggplot: (8757994532982)>

1.3.3 Significant correlation, FDR < 0.05

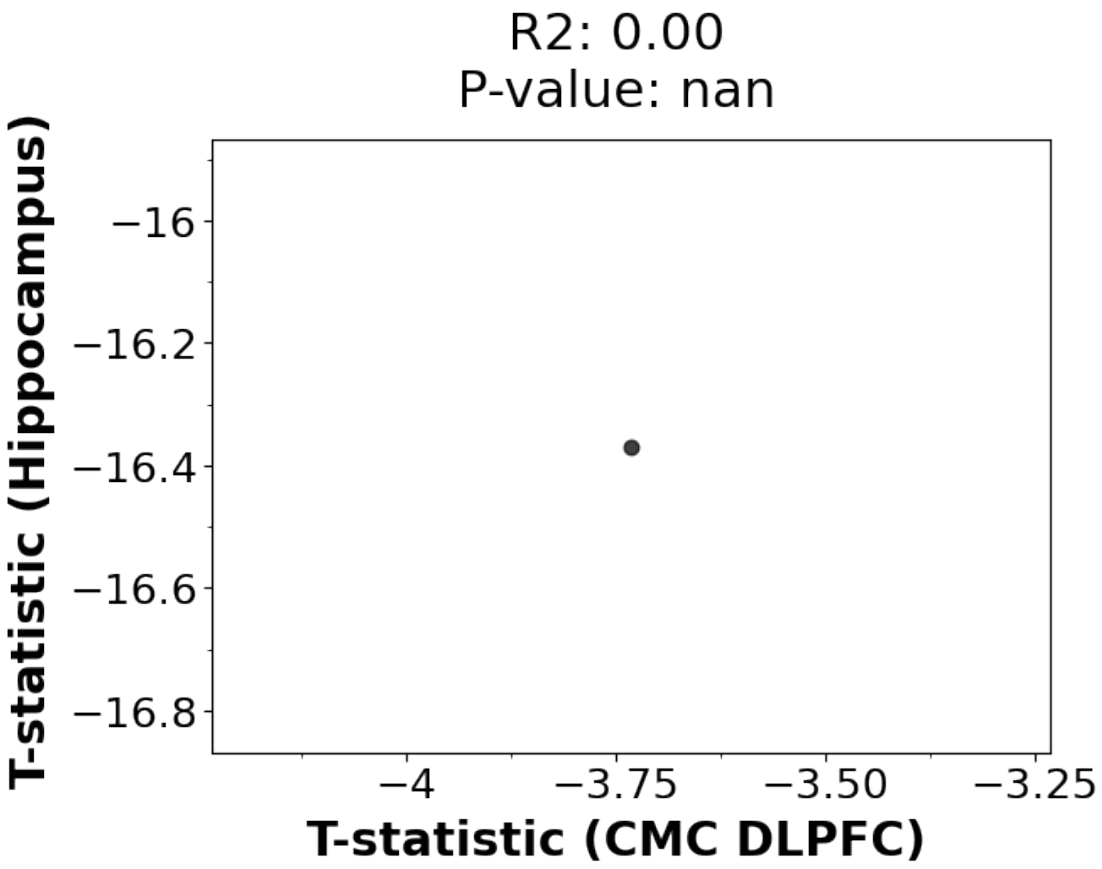
[40]:
```
pp = plot_corr('cmc_dlpfc', 'dlpfc', merge_dataframes_sig)
pp
```



[40]: <ggplot: (8757995377887)>

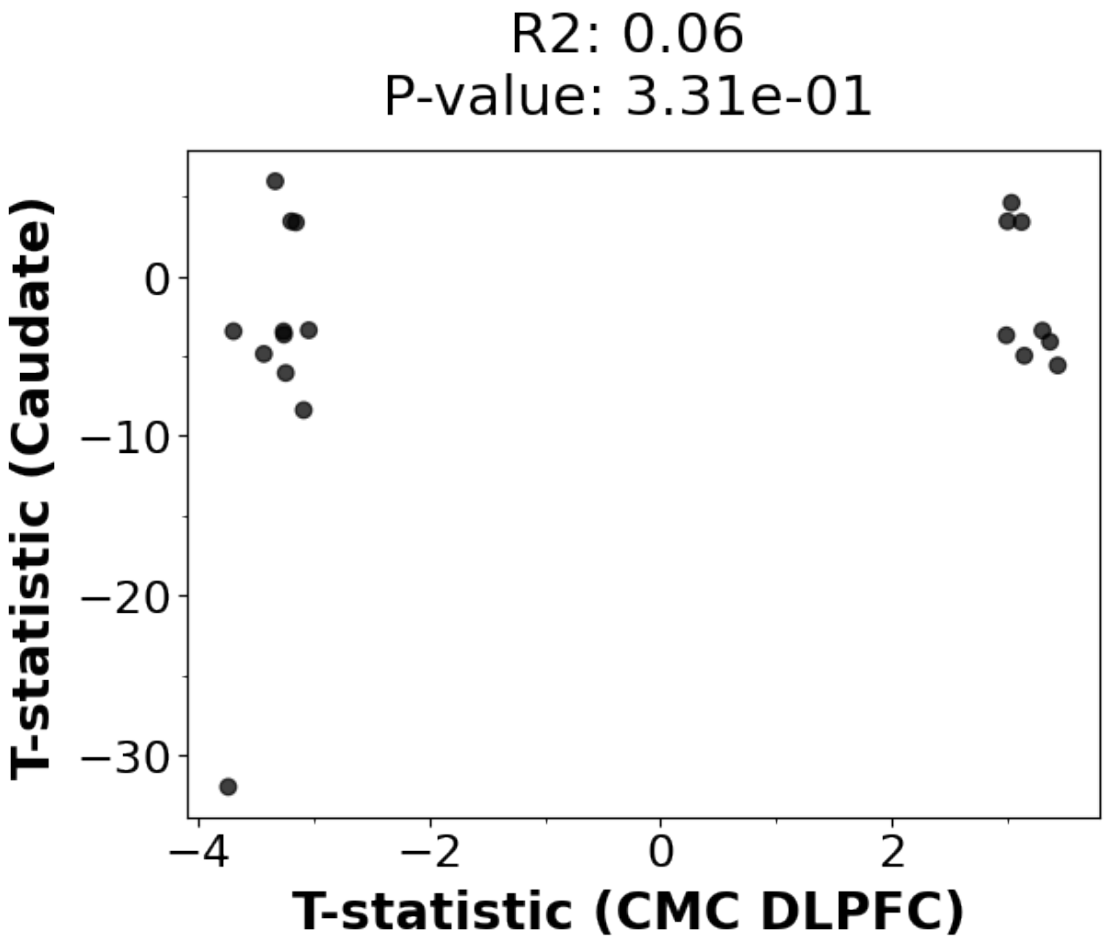
[41]:
```
qq = plot_corr('cmc_dlpfc', 'hippo', merge_dataframes_sig)
qq
```

```
/usr/lib/python3.9/site-packages/scipy/stats/_stats_mstats_common.py:160:
RuntimeWarning: invalid value encountered in double_scalars
/usr/lib/python3.9/site-packages/scipy/stats/_stats_mstats_common.py:174:
RuntimeWarning: invalid value encountered in sqrt
/usr/lib/python3.9/site-packages/scipy/stats/_stats_mstats_common.py:176:
RuntimeWarning: invalid value encountered in double_scalars
```



[41]:
```
<ggplot: (8757995385700)>
```

[42]:
```
ww = plot_corr('cmc_dlpfc', 'caudate', merge_dataframes_sig)
ww
```



[42]: <ggplot: (8757994704707)>

1.3.4 Directionality

All genes

[43]:
```
enrichment_binom('cmc_dlpfc', 'dlpfc', merge_dataframes)
```

```
   agree      0
0   -1.0   5417
1    1.0  12657
```

[43]: 5e-324

[44]:
```
enrichment_binom('cmc_dlpfc', 'hippo', merge_dataframes)
```

```
   agree     0
0   -1.0  8416
1    1.0  9554
```

[44]: 2.1653083863818587e-17

[45]:
```
enrichment_binom('cmc_dlpfc', 'caudate', merge_dataframes)
```

```
   agree     0
0   -1.0  9019
1    1.0  8861
```

[45]: 0.2403424427760989

Significant DEG (FDR < 0.05)

[46]:
```
enrichment_binom('cmc_dlpfc', 'dlpfc', merge_dataframes_sig)
```

```
   agree   0
0    1.0  77
All directions agree!
```

[47]:
```
enrichment_binom('cmc_dlpfc', 'hippo', merge_dataframes_sig)
```

```
   agree  0
0    1.0  1
All directions agree!
```

[48]:
```
enrichment_binom('cmc_dlpfc', 'caudate', merge_dataframes_sig)
```

```
   agree   0
0   -1.0   8
1    1.0  11
```

[48]: 0.6476058959960938

[49]:
```
df = merge_dataframes_sig("cmc_dlpfc", "caudate")
df[(df['agree']<0)]
```

[49]:

	Feature	ensemblID_cmc_dlpfc	adj.P.Val_cmc_dlpfc	\
2	ENSG00000130707.17	ENSG00000130707	0.033001	
4	ENSG00000055813.5	ENSG00000055813	0.034971	
5	ENSG00000225683.5	ENSG00000225683	0.035894	
6	ENSG00000185361.8	ENSG00000185361	0.036650	
10	ENSG00000141338.13	ENSG00000141338	0.040198	
11	ENSG00000135905.18	ENSG00000135905	0.041576	
12	ENSG00000050767.15	ENSG00000050767	0.041775	
18	ENSG00000174482.10	ENSG00000174482	0.048906	

	logFC_cmc_dlpfc	t_cmc_dlpfc	Dir_cmc_dlpfc	ensemblID_caudate	\
2	0.173408	3.427817	1.0	ENSG00000130707	
4	0.180729	3.361376	1.0	ENSG00000055813	
5	-0.324833	-3.326834	-1.0	ENSG00000225683	
6	0.138810	3.295714	1.0	ENSG00000185361	

10	-0.341710	-3.188087	-1.0	ENSG00000141338
11	-0.187448	-3.148755	-1.0	ENSG00000135905
12	0.177404	3.139331	1.0	ENSG00000050767
18	0.123179	2.984083	1.0	ENSG00000174482

	adj.P.Val_caudate	logFC_caudate	t_caudate	Dir_caudate	agree
2	0.000011	-0.272702	-5.581914	-1.0	-1.0
4	0.006252	-0.158076	-4.100884	-1.0	-1.0
5	0.000002	0.513122	5.959243	1.0	-1.0
6	0.046721	-0.113374	-3.402620	-1.0	-1.0
10	0.042255	0.183900	3.447082	1.0	-1.0
11	0.049758	0.124795	3.379555	1.0	-1.0
12	0.000205	-0.336119	-4.974989	-1.0	-1.0
18	0.021891	-0.160731	-3.696314	-1.0	-1.0

[ ]: