main

July 9, 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': '../../../_m/genes/diffExpr_szVctl_full.txt',
    'dlpfc': '/ceph/projects/v4_phase3_paper/inputs/public_data/_m/phase2/
 ↪dlpfc_diffExpr_szVctl_full.txt',
    'hippo': '/ceph/projects/v4_phase3_paper/inputs/public_data/_m/phase2/
 ↪hippo_diffExpr_szVctl_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):
    fdr1 = 0.05 if tissue1 != 'dlpfc' else 0.05
    fdr2 = 0.05 if tissue2 != 'dlpfc' else 0.05
    return get_deg_sig(config[tissue1], fdr1).
 ↪merge(get_deg_sig(config[tissue2], fdr2),
                                                  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)
    fdr1 = 0.05 if tissue1 != 'dlpfc' else 0.05
    fdr2 = 0.05 if tissue2 != 'dlpfc' else 0.05
    table = [[np.sum((df['adj.P.Val_%s' % tissue1]<fdr1) &
                     ((df['adj.P.Val_%s' % tissue2]<fdr2))),
              np.sum((df['adj.P.Val_%s' % tissue1]<fdr1) &
                     ((df['adj.P.Val_%s' % tissue2]>=fdr2)))],
             [np.sum((df['adj.P.Val_%s' % tissue1]>=fdr1) &
                     ((df['adj.P.Val_%s' % tissue2]<fdr2))),
              np.sum((df['adj.P.Val_%s' % tissue1]>=fdr1) &
                     ((df['adj.P.Val_%s' % tissue2]>=fdr2)))]]
    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'}[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/v4_phase3_paper/inputs/phenotypes/_m/
 ↪merged_phenotypes.csv'
pheno = pd.read_csv(pheno_file, index_col=0)
pheno = pheno[(pheno['Age'] > 17) &
              (pheno['Dx'].isin(['SZ', 'CTL'])) &
              (pheno['Race'].isin(['AA', "EA"]))].copy()
pheno.head(2)
```

[6]:

RNum	Sex	Race	Dx	Age	mitoRate	rRNA_rate	totalAssignedGene	RIN	\
R11135	Male	EA	CTL	18.77	0.257280	0.000169	0.523132	5.9	
R11137	Male	EA	CTL	41.44	0.384027	0.000088	0.593343	9.2	

RNum	ERCCsumLogErr	overallMapRate	snpPC1	snpPC2	snpPC3	snpPC4	\
R11135	-22.049787	0.8746	-0.036163	0.003232	0.000562	0.001725	
R11137	-29.498329	0.9149	-0.035985	0.003539	-0.000170	-0.001330	

RNum	snpPC5	Region	BrNum	antipsychotics	lifetime_antipsych	Protocol
R11135	-0.000807	HIPPO	Br2063	False	False	RiboZeroHMR
R11137	0.002003	HIPPO	Br2582	False	False	RiboZeroHMR

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

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

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

[8]:
```
Region   Race
Caudate  AA      205
         EA      189
DLPFC    AA      200
         EA      160
HIPPO    AA      207
         EA      169
```

```
dtype: int64
```

[9]:
```
pheno.groupby(['Region', 'Race', 'Sex']).size()
```

[9]:
```
Region   Race  Sex
Caudate  AA    Female     78
               Male      127
         EA    Female     43
               Male      146
DLPFC    AA    Female     75
               Male      125
         EA    Female     39
               Male      121
HIPPO    AA    Female     81
               Male      126
         EA    Female     40
               Male      129
dtype: int64
```

1.2 BrainSeq Tissue Comparison

[10]:
```
caudate = get_deg(config['caudate'])
caudate.groupby('Dir').size()
```

[10]:
```
Dir
-1.0    12061
 1.0    10897
dtype: int64
```

[11]:
```
caudate[(caudate['adj.P.Val'] < 0.05)].shape
```

[11]:
```
(2701, 6)
```

[12]:
```
dlpfc = get_deg(config['dlpfc'])
dlpfc.groupby('Dir').size()
```

[12]:
```
Dir
-1.0    13207
 1.0    11445
dtype: int64
```

[13]:
```
dlpfc[(dlpfc['adj.P.Val'] < 0.05)].shape
```

[13]:
```
(245, 6)
```

[14]:
```
hippo = get_deg(config['hippo'])
hippo.groupby('Dir').size()
```

[14]:
```
Dir
-1.0    12852
 1.0    11800
dtype: int64
```

[15]:
```
hippo[(hippo['adj.P.Val'] < 0.05)].shape
```

[15]:
```
(48, 6)
```

1.2.1 Enrichment of DEG

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

```
[[49, 2498], [180, 18132]]
```

[16]:
```
(1.975954096610622, 9.40458506586896e-05)
```

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

```
[[10, 2537], [35, 18277]]
```

[17]:
```
(2.0583366180528184, 0.06245006401479434)
```

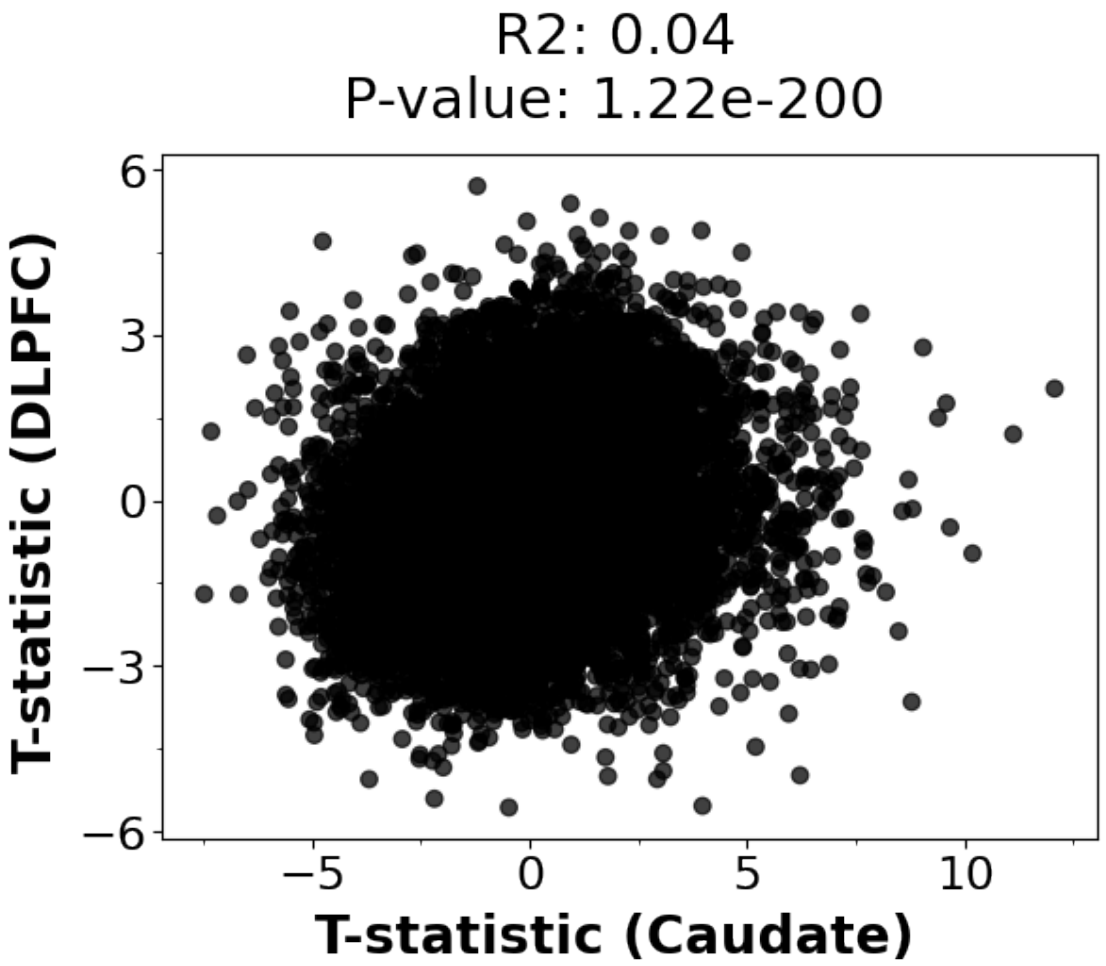
[18]:
```
cal_fishers('dlpfc', 'hippo')
```

```
[[6, 239], [42, 24365]]
```

[18]:
```
(14.563658099222954, 7.842543158014382e-06)
```

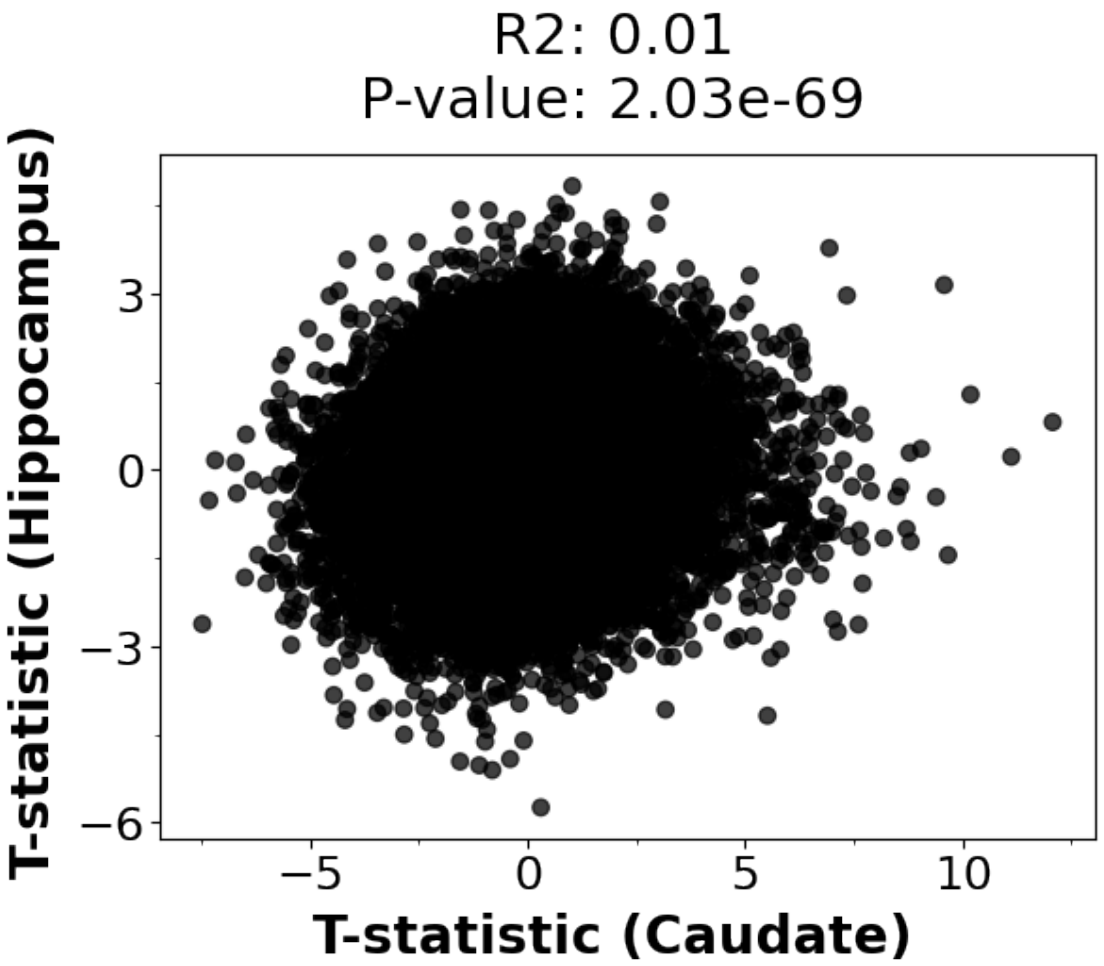
1.2.2 Correlation

[19]:
```
pp = plot_corr('caudate', 'dlpfc', merge_dataframes)
pp
```



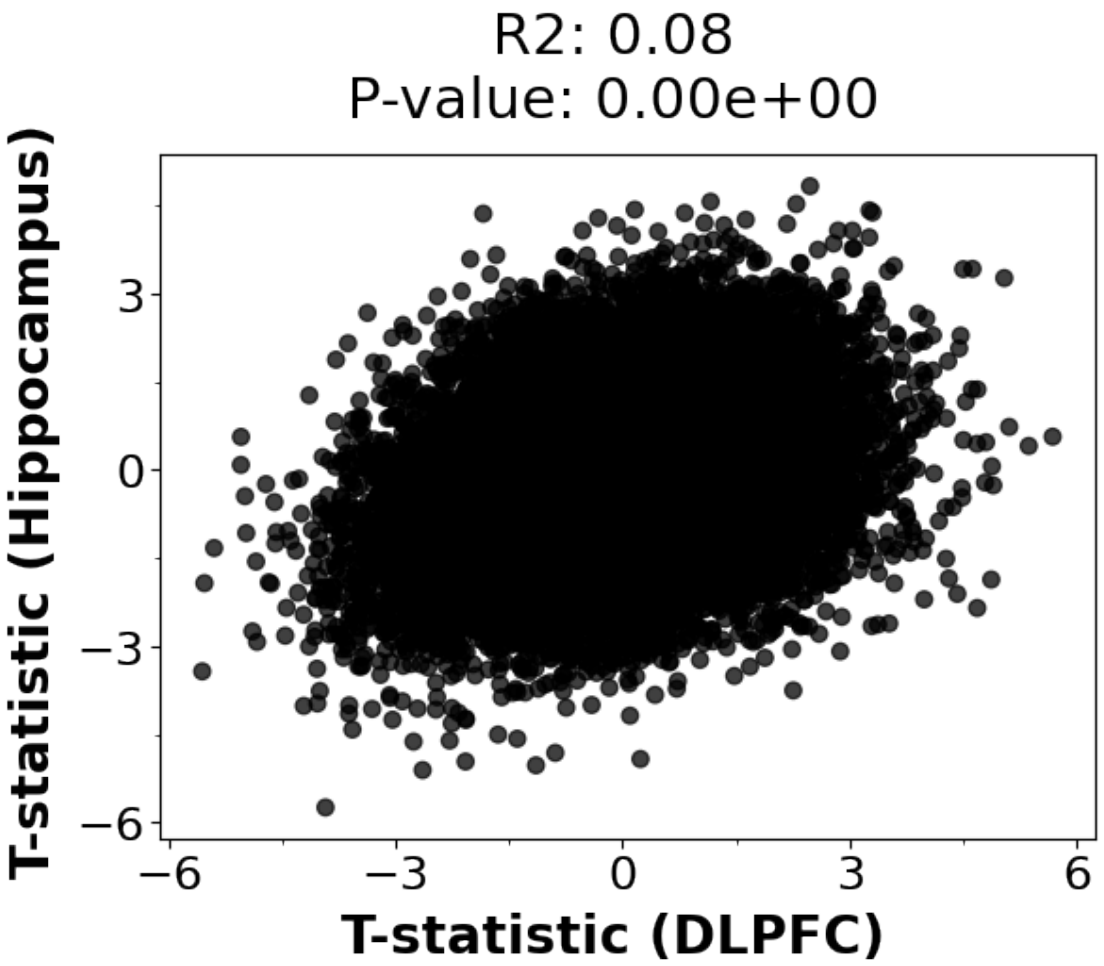
[19]: <ggplot: (8736556688119)>

[20]:
```
qq = plot_corr('caudate', 'hippo', merge_dataframes)
qq
```



[20]: <ggplot: (8736556376480)>

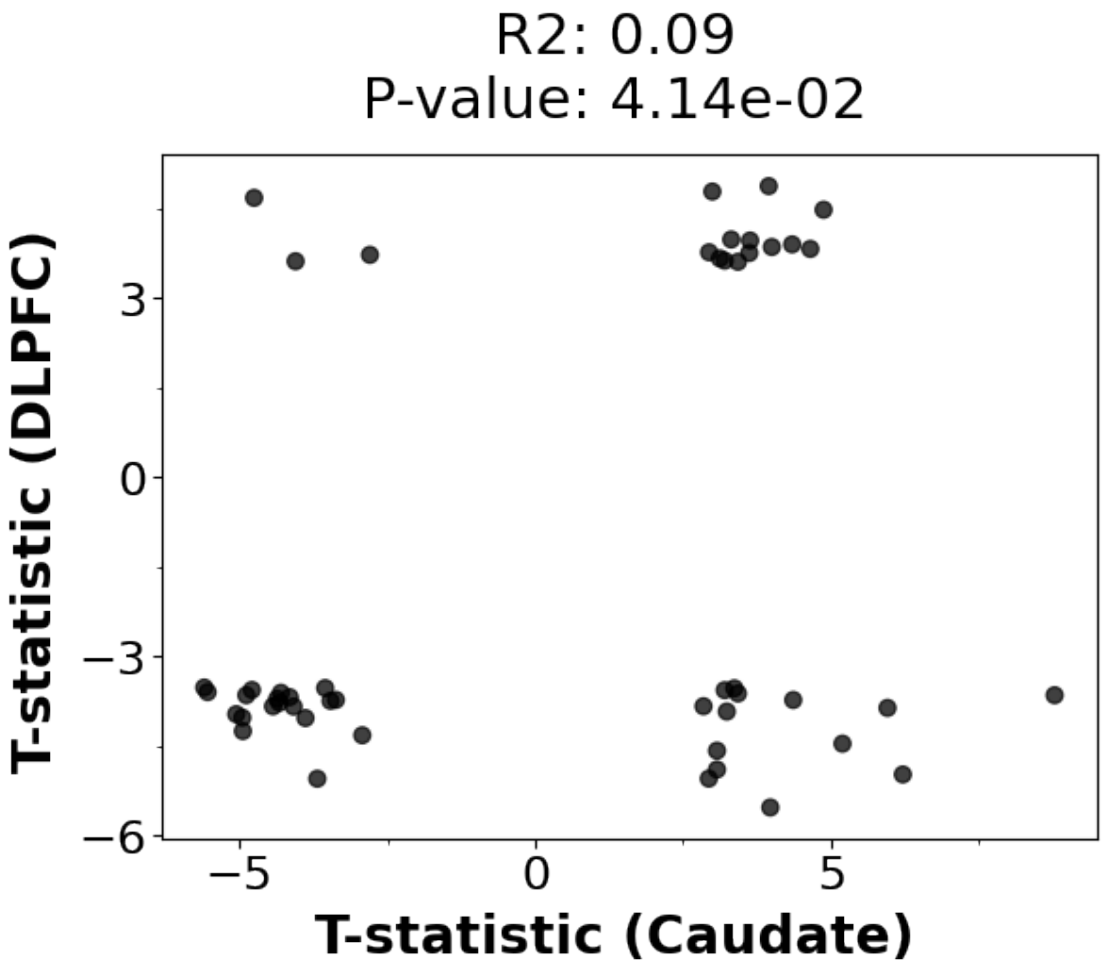
[21]:
```
ww = plot_corr('dlpfc', 'hippo', merge_dataframes)
ww
```



[21]: <ggplot: (8736556379796)>

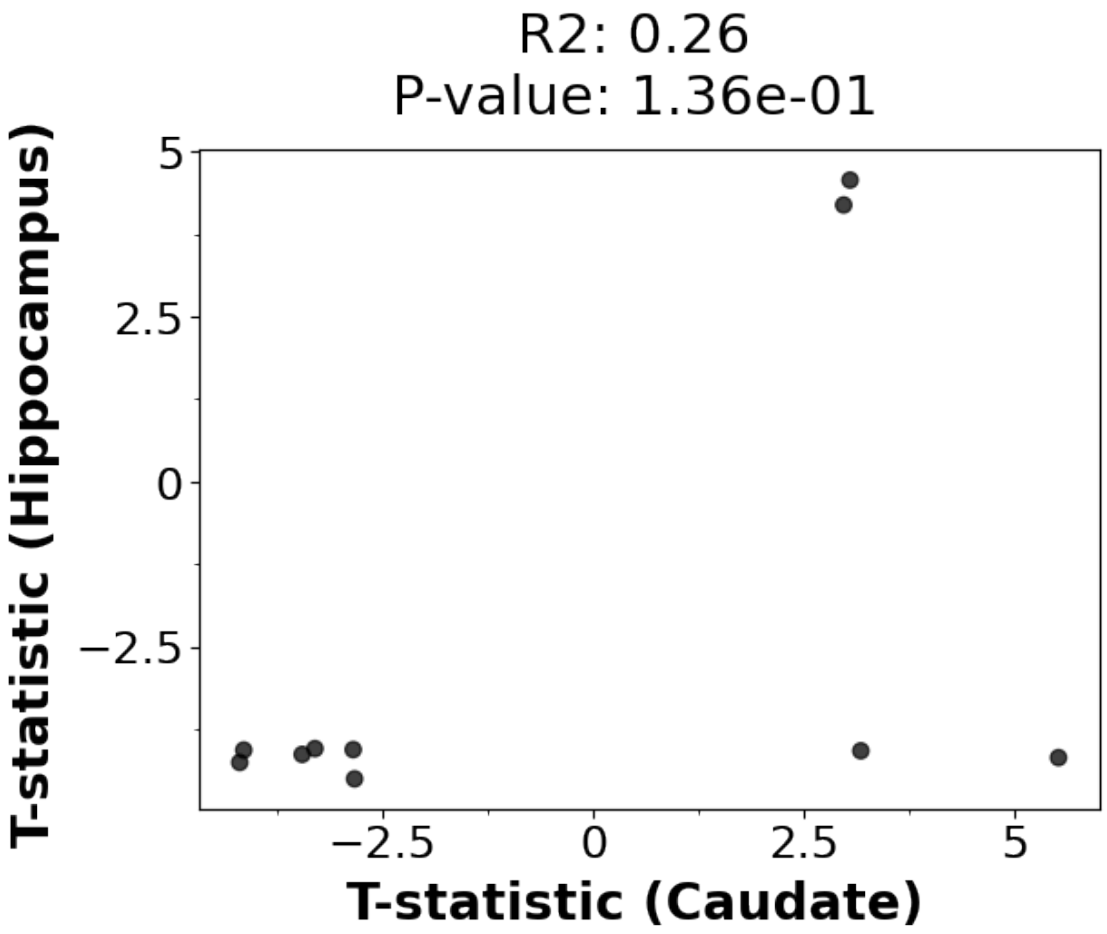
1.2.3 Significant correlation, FDR < 0.05

[22]:
```
pp = plot_corr('caudate', 'dlpfc', merge_dataframes_sig)
pp
```



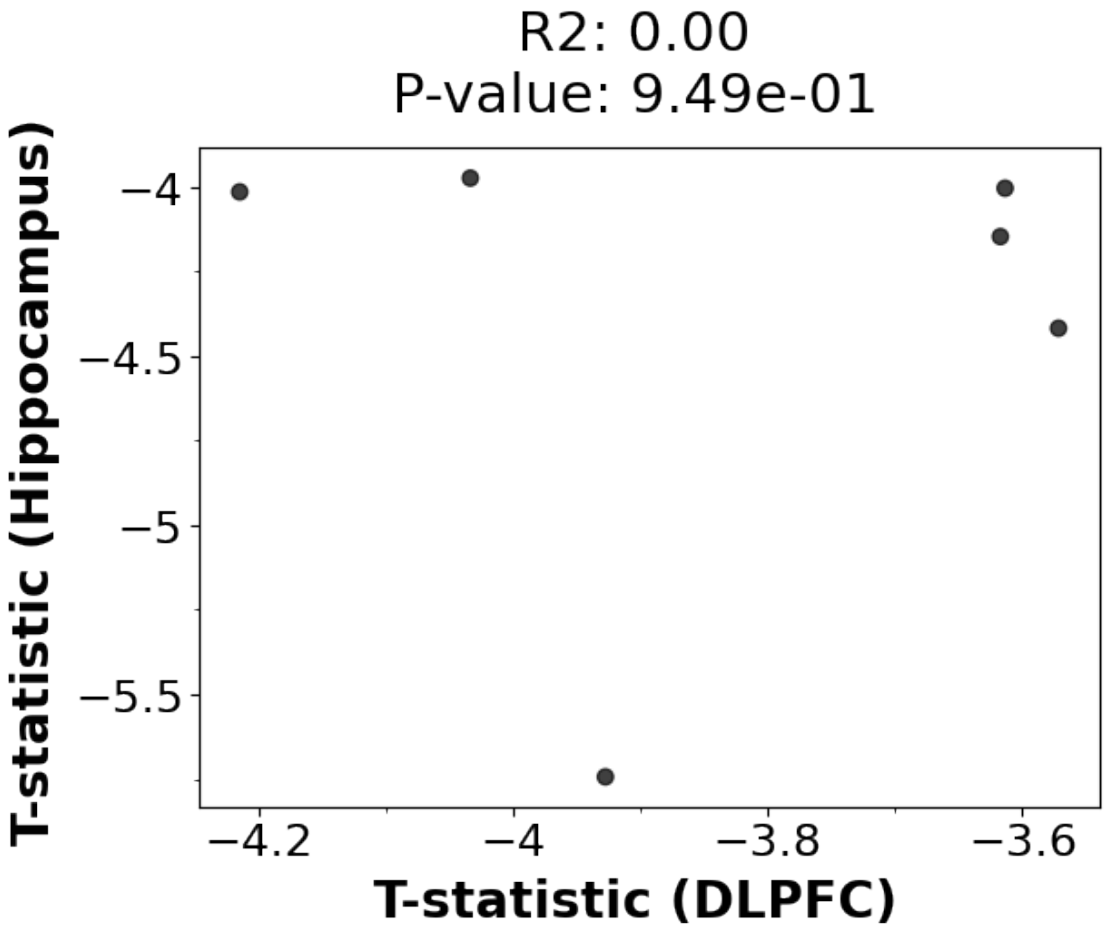
[22]: <ggplot: (8736551736808)>

[23]:
```
qq = plot_corr('caudate', 'hippo', merge_dataframes_sig)
qq
```



[23]: <ggplot: (8736556369966)>

[24]:
```
ww = plot_corr('dlpfc', 'hippo', merge_dataframes_sig)
ww
```



[24]: <ggplot: (8736551779193)>

1.2.4 Directionality test

All genes

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

	agree	0
0	-1.0	8821
1	1.0	12038

[25]: 2.8390706399014226e-110

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

	agree	0
0	-1.0	9545
1	1.0	11314

[26]: 1.7045050229182753e-34

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

```
   agree      0
0   -1.0  10291
1    1.0  14361
```

[27]: 9.504008229727351e-149

Significant DEG (FDR < 0.05)

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

```
   agree   0
0   -1.0  17
1    1.0  32
```

[28]: 0.04438416098714981

[29]:
```
enrichment_binom('caudate', 'hippo', merge_dataframes_sig)
```

```
   agree  0
0   -1.0  2
1    1.0  8
```

[29]: 0.10937500000000003

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

```
   agree  0
0    1.0  6
All directions agree!
```

[ ]: