

main

July 14, 2021

1 Visualize GO analysis

```
[1]: import numpy as np
import pandas as pd
```

```
[2]: def get_top_GO(tissue, fn, label):
    df = pd.read_csv(fn, sep='\t').sort_values('p_value').head(10)
    df['Log10'] = -np.log10(df['p_value'])
    df['Tissue'] = tissue
    df['Direction'] = label
    return df
```

```
[3]: tissue = 'hippocampus'
config = {
    'All': '../_m/DEGs_functional_enrichment.tsv',
    'Up': '../_m/upreg_DEGs_functional_enrichment.tsv',
    'Down': '../_m/downreg_DEGs_functional_enrichment.tsv',
}

df = pd.DataFrame()
for bias in ['All', 'Up', 'Down']:
    df = pd.concat([df, get_top_GO(tissue, config[bias], bias)], axis=0)

df.shape
```

```
[3]: (30, 17)
```

```
[4]: df.to_csv("%s_functional_analysis.tsv" % tissue, sep='\t', index=False)
```

1.1 Plot

```
[5]: %load_ext rpy2.ipython
```

```
[6]: %%R -i df
library(ggplot2)
library(tidyverse)

save_plot <- function(p, fn, w, h){
```

```

    for(ext in c('.svg', '.png', '.pdf')){
      ggsave(file=paste0(fn,ext), plot=p, width=w, height=h)
    }
  }

plot_G0 <- function(){
  cbPalette <- c("#000000", "Red", "Blue")
  gg1 = df %>%
    ggplot(aes(x=Log10, y=term_name, color=Direction)) +
    geom_point(shape=18, alpha=0.8, size=4) + labs(y='', x='-Log10 (p
    ↪adjust)') +
    theme_bw() +
    scale_colour_manual(name="Direction", values=cbPalette,
      labels=c("All", "Upregulated in SZ", "Downregulated
    ↪in SZ")) +
    geom_vline(xintercept = -log10(0.05), linetype = "dotted") +
    theme(axis.text=element_text(size=14),
      axis.title=element_text(size=18, face='bold'),
      strip.text=element_text(size=18, face='bold'))
  return(gg1)
}

```

```

R[write to console]: Attaching packages
                     tidyverse 1.3.1

```

```

R[write to console]:  tidbale  3.1.2      dplyr    1.0.7
                     tidyr    1.1.3      stringr  1.4.0
                     readr    1.4.0      forcats  0.5.1
                     purrr    0.3.4

```

```

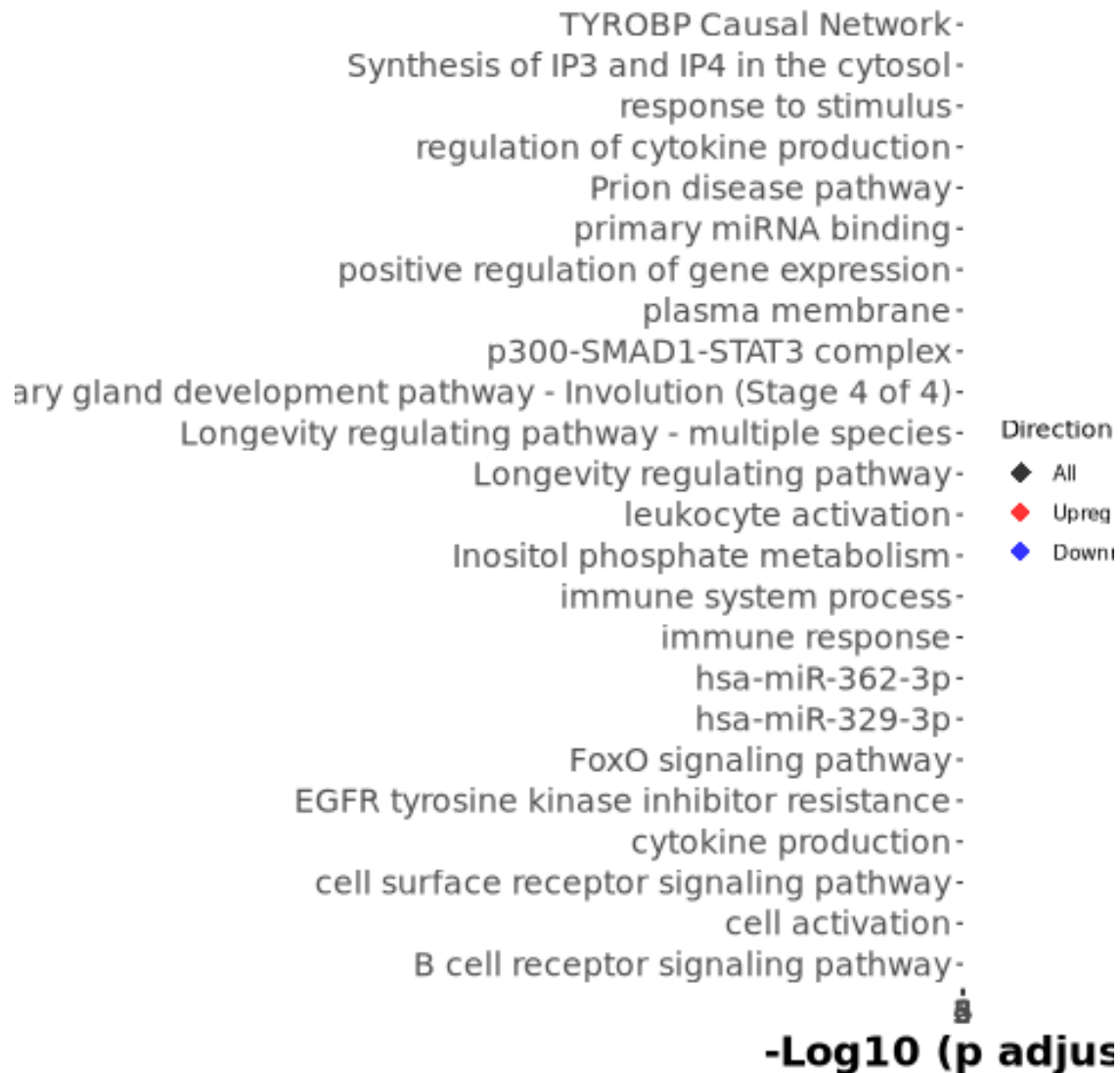
R[write to console]: Conflicts
tidyverse_conflicts()
dplyr::filter() masks stats::filter()
dplyr::lag()    masks stats::lag()

```

```

[7]: %>%R
gg1 = plot_G0()
print(gg1)
save_plot(gg1, "hippocampus_G0_top10_stacked", 10, 6)

```



[]: