

main

February 3, 2022

1 Examining the overlap between antipsychotics analysis with main DE analysis

```
[1]: import pandas as pd
import matplotlib.pyplot as plt
from matplotlib_venn import venn2,venn3
```

1.1 Load data

1.1.1 Load DE genes results: BrainSEQ caudate

```
[2]: def get_degs(df, fdr):
    return df[(df["adj.P.Val"] < fdr)]

ap = get_degs(pd.read_csv("../_m/genes/diffExpr_sz_APVctl_full.txt",
    ↪sep='\t', index_col=0), 0.05)
noap = get_degs(pd.read_csv("../_m/genes/diffExpr_sz_noAPVctl_full.txt",
    ↪sep='\t', index_col=0), 0.05)
degs = get_degs(pd.read_csv("../_m/genes/diffExpr_szVctl_full.txt",
    ↪sep='\t', index_col=0), 0.05)
```

1.1.2 Load animal study results

```
[3]: def mouse2human():
    fn = "/ceph/users/jbenja13/projects/aanri/racial_diff/input/celltypes/_h/
    ↪cell_type/mouse2human_gene.txt"
    return pd.read_csv(fn, sep='\t')

chong_2002 = pd.read_csv("../_m/Chong2002_rat.csv")
kim_2018 = pd.read_csv("../_m/Kim2018_mice.csv")\
    .merge(mouse2human(), left_on="Gene symbol",
    ↪right_on="Symbol_mouse")
kim_2018 = kim_2018[(kim_2018["Tissue"] == "Striatum")].copy()
korostynski_2013 = pd.read_csv("../_m/Korostynski2013_mice.csv")\
    .merge(mouse2human(), left_on="Gene_symbol",
    ↪right_on="Symbol_mouse")
```

```

korostynski_2013 = korostynski_2013[(korostynski_2013["ANOVA_drug_FDR"] < 0.
→05)]\
    .groupby("Gene_symbol").first().reset_index().
→sort_values("ANOVA_drug_FDR")

```

1.2 Study overlap

1.2.1 Percentage of overlap

SZ vs control

```

[4]: a = len(set(degs.Symbol) & set(chong_2002.loc[:, "Gene Name (Human)"]))
b = a / len(set(chong_2002.loc[:, "Gene Name (Human)"]))
c = a / len(set(degs.Symbol))
print("Chong 2002 total DE: %d\n" % len(set(chong_2002.loc[:, "Gene Name_
→(Human)"])))
print("Overlap with Chong 2002 (#): %d\nPercent Overlap (Chong): %.
→1f%%\nPercent Overlap (BrainSEQ): %.1f%%" % (a, b*100, c*100))

```

Chong 2002 total DE: 14

Overlap with Chong 2002 (#): 6
Percent Overlap (Chong): 42.9%
Percent Overlap (BrainSEQ): 0.2%

```

[5]: a = len(set(degs.Symbol) & set(korostynski_2013.Symbol_human))
b = a / len(set(korostynski_2013.Symbol_human))
c = a / len(set(degs.Symbol))
print("Korostynski 2013 total DE: %d\n" % len(set(korostynski_2013.
→Symbol_human)))
print("Overlap with Korostynski 2013 (#): %d\nPercent Overlap (Korostynski): %.
→1f%%\nPercent Overlap (BrainSEQ): %.1f%%" % (a, b*100, c*100))

```

Korostynski 2013 total DE: 642

Overlap with Korostynski 2013 (#): 99
Percent Overlap (Korostynski): 15.4%
Percent Overlap (BrainSEQ): 4.0%

```

[6]: a = len(set(degs.Symbol) & set(kim_2018.Symbol_human))
b = a / len(set(kim_2018.Symbol_human))
c = a / len(set(degs.Symbol))
print("Kim 2018 total DE: %d\n" % len(set(kim_2018.Symbol_human)))
print("Overlap with Kim 2018 (#): %d\nPercent Overlap (Kim): %.1f%%\nPercent_
→Overlap (BrainSEQ): %.1f%%" % (a, b*100, c*100))

```

Kim 2018 total DE: 59

Overlap with Kim 2018 (#): 15
Percent Overlap (Kim): 25.4%

Percent Overlap (BrainSEQ): 0.6%

SZ AP vs control

```
[7]: a = len(set(ap.Symbol) & set(chong_2002.loc[:, "Gene Name (Human)"]))
b = a / len(set(chong_2002.loc[:, "Gene Name (Human)"]))
c = a / len(set(ap.Symbol))
print("Chong 2002 total DE: %d\n" % len(set(chong_2002.loc[:, "Gene Name_
→(Human)"])))
print("Overlap with Chong 2002 (#): %d\nPercent Overlap (Chong): %.
→1f%\nPercent Overlap (BrainSEQ): %.1f%" % (a, b*100, c*100))
```

Chong 2002 total DE: 14

Overlap with Chong 2002 (#): 6

Percent Overlap (Chong): 42.9%

Percent Overlap (BrainSEQ): 0.2%

```
[8]: a = len(set(ap.Symbol) & set(korostynski_2013.Symbol_human))
b = a / len(set(korostynski_2013.Symbol_human))
c = a / len(set(ap.Symbol))
print("Korostynski 2013 total DE: %d\n" % len(set(korostynski_2013.
→Symbol_human)))
print("Overlap with Korostynski 2013 (#): %d\nPercent Overlap (Korostynski): %.
→1f%\nPercent Overlap (BrainSEQ): %.1f%" % (a, b*100, c*100))
```

Korostynski 2013 total DE: 642

Overlap with Korostynski 2013 (#): 98

Percent Overlap (Korostynski): 15.3%

Percent Overlap (BrainSEQ): 4.0%

```
[9]: a = len(set(ap.Symbol) & set(kim_2018.Symbol_human))
b = a / len(set(kim_2018.Symbol_human))
c = a / len(set(ap.Symbol))
print("Kim 2018 total DE: %d\n" % len(set(kim_2018.Symbol_human)))
print("Overlap with Kim 2018 (#): %d\nPercent Overlap (Kim): %.1f%\nPercent_
→Overlap (BrainSEQ): %.1f%" % (a, b*100, c*100))
```

Kim 2018 total DE: 59

Overlap with Kim 2018 (#): 14

Percent Overlap (Kim): 23.7%

Percent Overlap (BrainSEQ): 0.6%

SZ noAP vs control

```
[10]: a = len(set(noap.Symbol) & set(chong_2002.loc[:, "Gene Name (Human)"]))
b = a / len(set(chong_2002.loc[:, "Gene Name (Human)"]))
c = a / len(set(noap.Symbol))
```

```
print("Chong 2002 total DE: %d\n" % len(set(chong_2002.loc[:, "Gene Name_
→(Human)"])))
print("Overlap with Chong 2002 (#): %d\nPercent Overlap (Chong): %.
→1f%%\nPercent Overlap (BrainSEQ): %.1f%%" % (a, b*100, c*100))
```

Chong 2002 total DE: 14

Overlap with Chong 2002 (#): 1
Percent Overlap (Chong): 7.1%
Percent Overlap (BrainSEQ): 0.2%

```
[11]: a = len(set(noap.Symbol) & set(korostynski_2013.Symbol_human))
b = a / len(set(korostynski_2013.Symbol_human))
c = a / len(set(noap.Symbol))
print("Korostynski 2013 total DE: %d\n" % len(set(korostynski_2013.
→Symbol_human)))
print("Overlap with Korostynski 2013 (#): %d\nPercent Overlap (Korostynski): %.
→1f%%\nPercent Overlap (BrainSEQ): %.1f%%" % (a, b*100, c*100))
```

Korostynski 2013 total DE: 642

Overlap with Korostynski 2013 (#): 36
Percent Overlap (Korostynski): 5.6%
Percent Overlap (BrainSEQ): 5.8%

```
[12]: a = len(set(noap.Symbol) & set(kim_2018.Symbol_human))
b = a / len(set(kim_2018.Symbol_human))
c = a / len(set(noap.Symbol))
print("Kim 2018 total DE: %d\n" % len(set(kim_2018.Symbol_human)))
print("Overlap with Kim 2018 (#): %d\nPercent Overlap (Kim): %.1f%%\nPercent_
→Overlap (BrainSEQ): %.1f%%" % (a, b*100, c*100))
```

Kim 2018 total DE: 59

Overlap with Kim 2018 (#): 3
Percent Overlap (Kim): 5.1%
Percent Overlap (BrainSEQ): 0.5%

1.2.2 Plot venn diagrams

```
[13]: plt.rcParams.update({'font.size': 22, 'font.weight': 'bold'})
```

```
[14]: def plot_pairwise_venn(gene_names, dfx, label1, label2):
    fn = "venn_%s_%s" % (label1.replace(" ", "_"), label2.replace(" ", "_"))
    plt.figure(figsize=(8,8))
    v = venn2([gene_names, set(dfx.Symbol)],
               set_labels = (label1, "BrainSEQ (%s)"%label2))
    v.get_patch_by_id('10').set_color('red')
```

```

v.get_patch_by_id('10').set_alpha(0.75)
v.get_patch_by_id('01').set_color('blue')
v.get_patch_by_id('01').set_alpha(0.75)
try:
    v.get_patch_by_id('11').set_color('purple')
    v.get_patch_by_id('11').set_alpha(0.75)
except AttributeError:
    print("There is no overlap!")
plt.savefig('%s.png' % fn)
plt.savefig('%s.pdf' % fn)

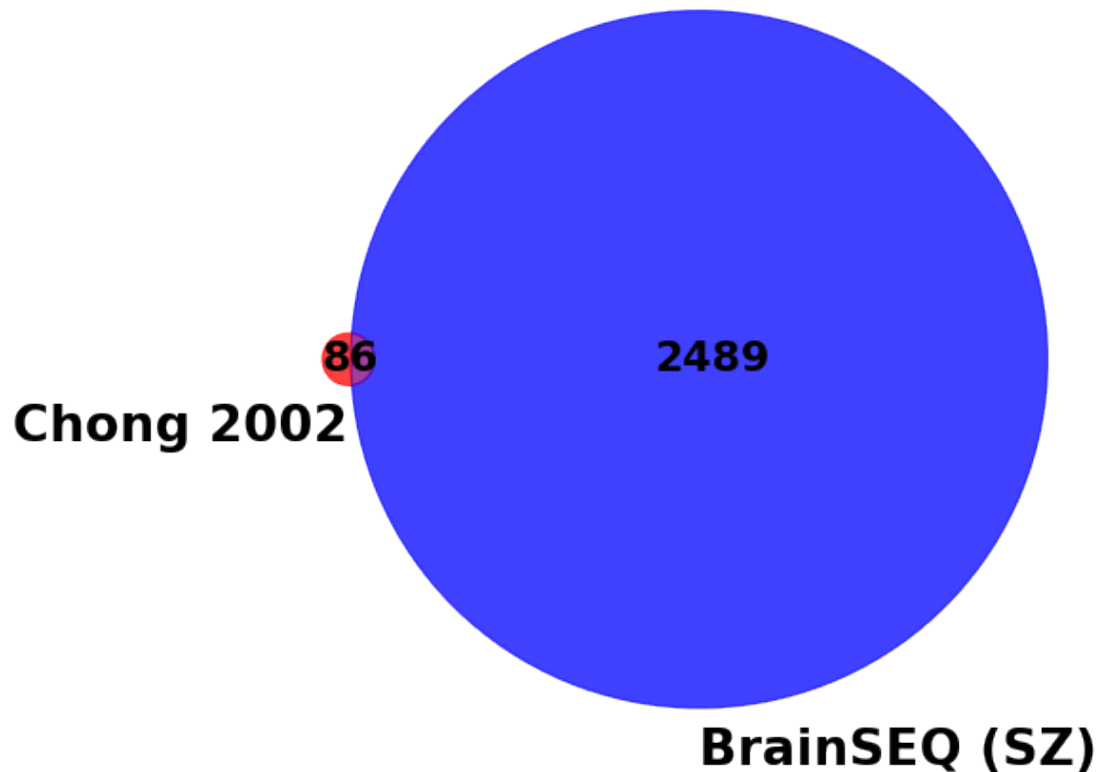
```

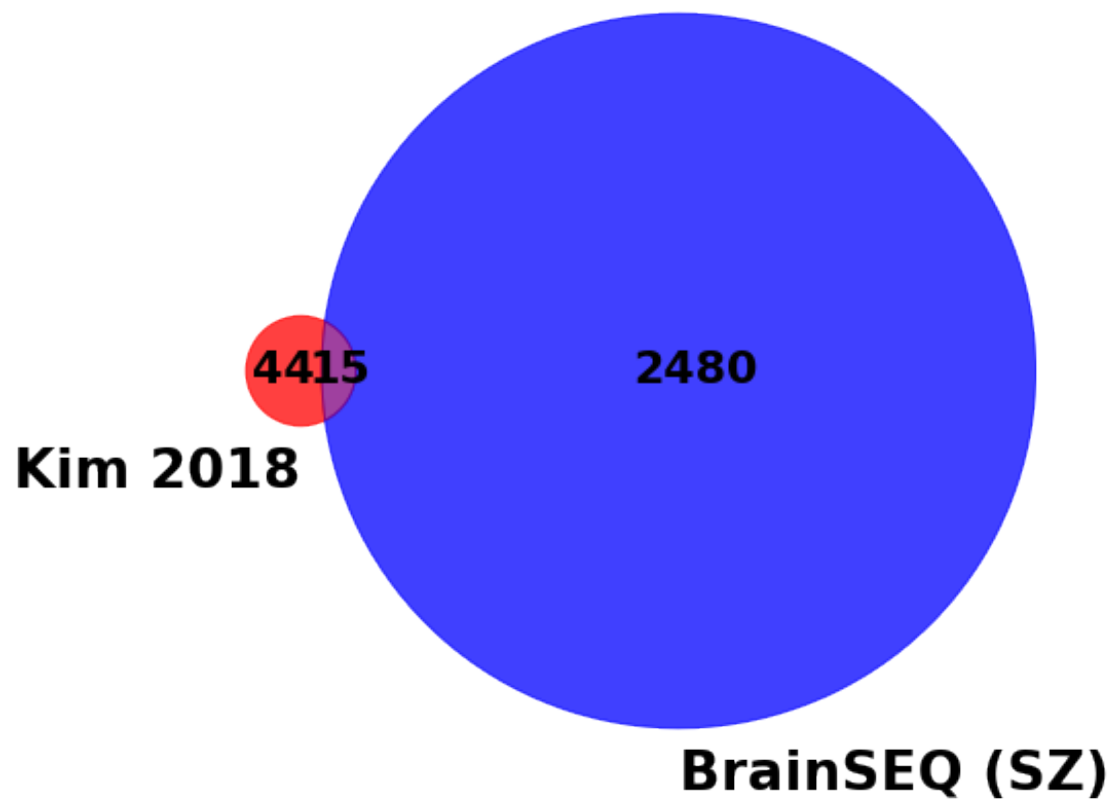
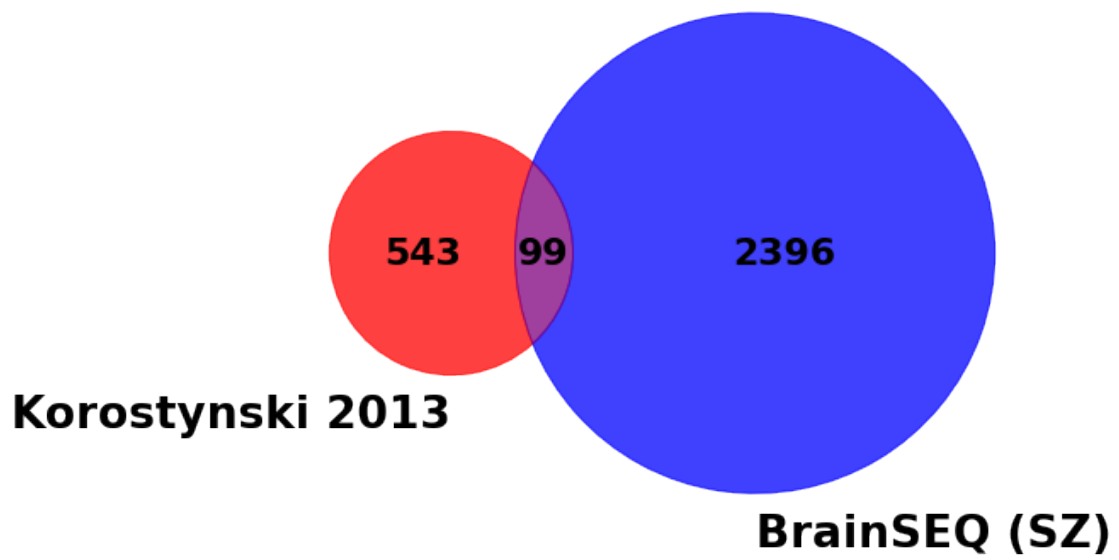
SZ vs control

```

[15]: plot_pairwise_venn(set(chong_2002.loc[:, "Gene Name (Human)"]), degs, "Chong_
    ↪2002", "SZ")
plot_pairwise_venn(set(korostynski_2013.Symbol_human), degs, "Korostynski_
    ↪2013", "SZ")
plot_pairwise_venn(set(kim_2018.Symbol_human), degs, "Kim 2018", "SZ")

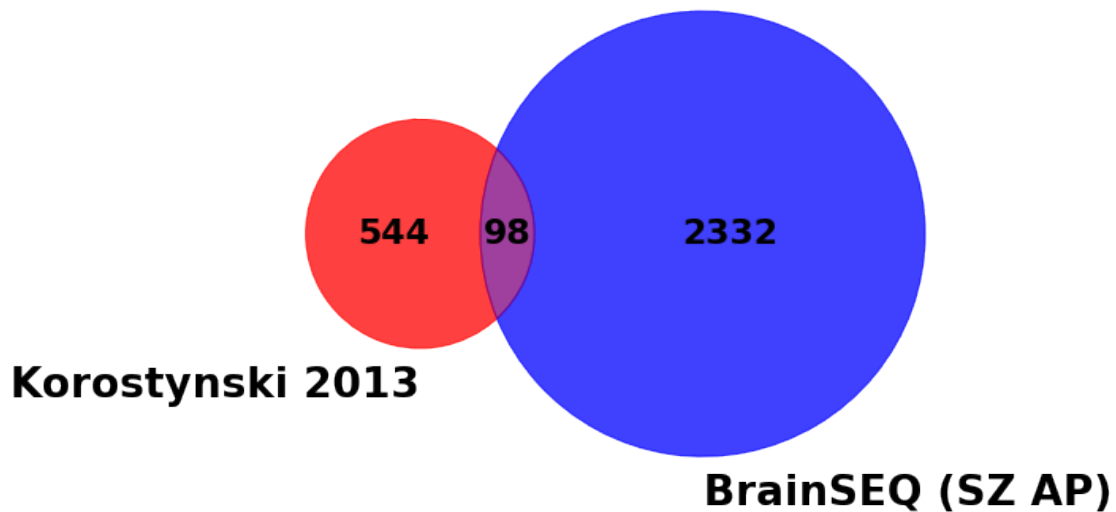
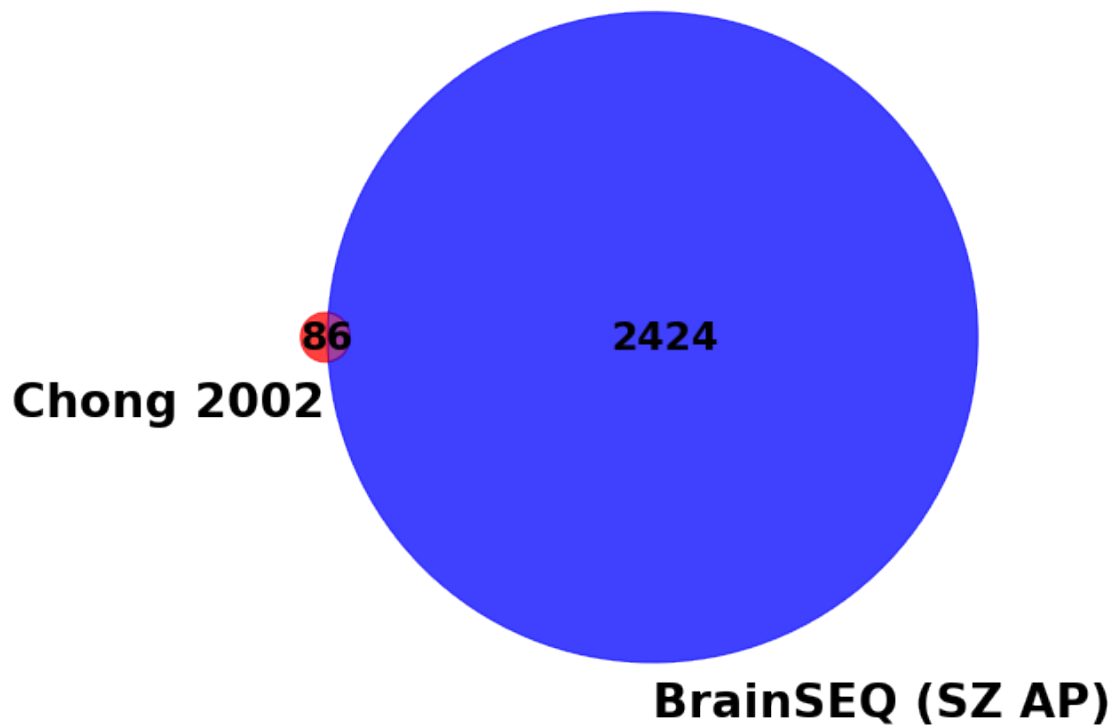
```

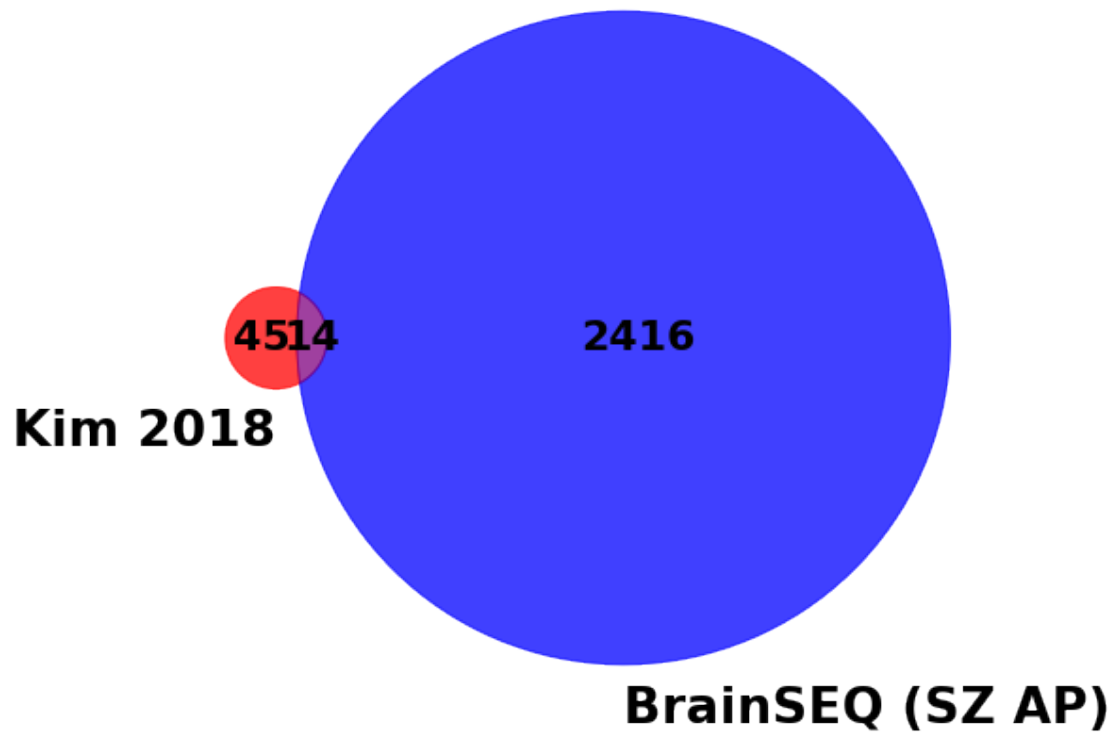




SZ AP vs control

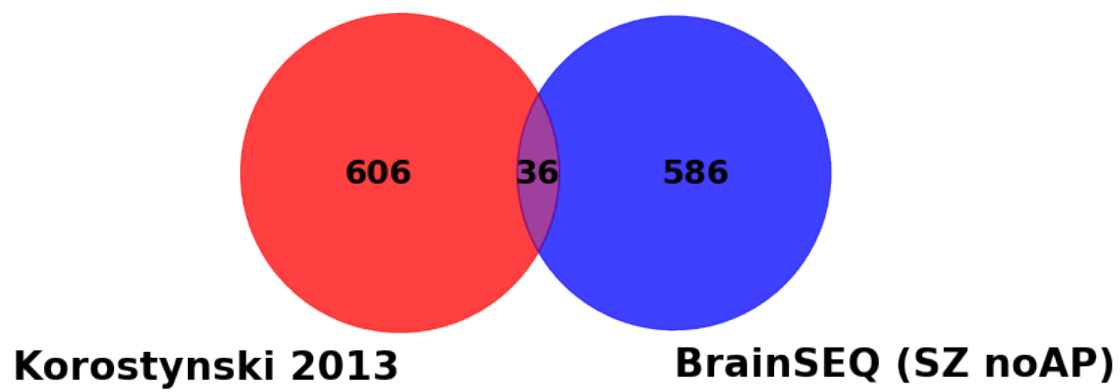
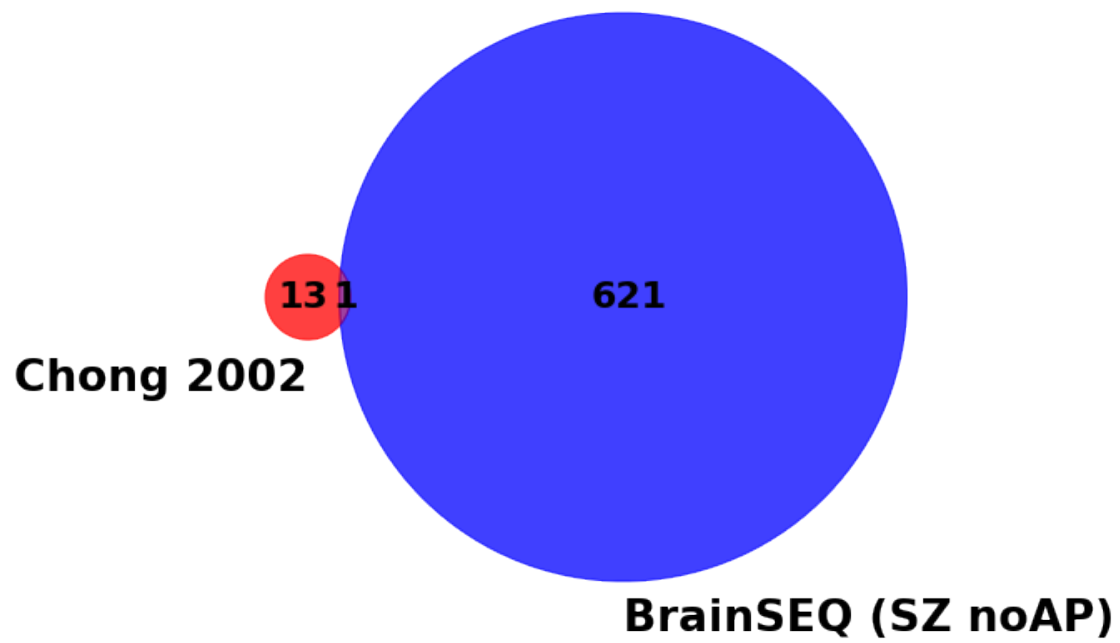
```
[16]: plot_pairwise_venn(set(chong_2002.loc[:, "Gene Name (Human)"]), ap, "Chong_2002", "SZ AP")
      plot_pairwise_venn(set(korostynski_2013.Symbol_human), ap, "Korostynski 2013", "SZ AP")
      plot_pairwise_venn(set(kim_2018.Symbol_human), ap, "Kim 2018", "SZ AP")
```

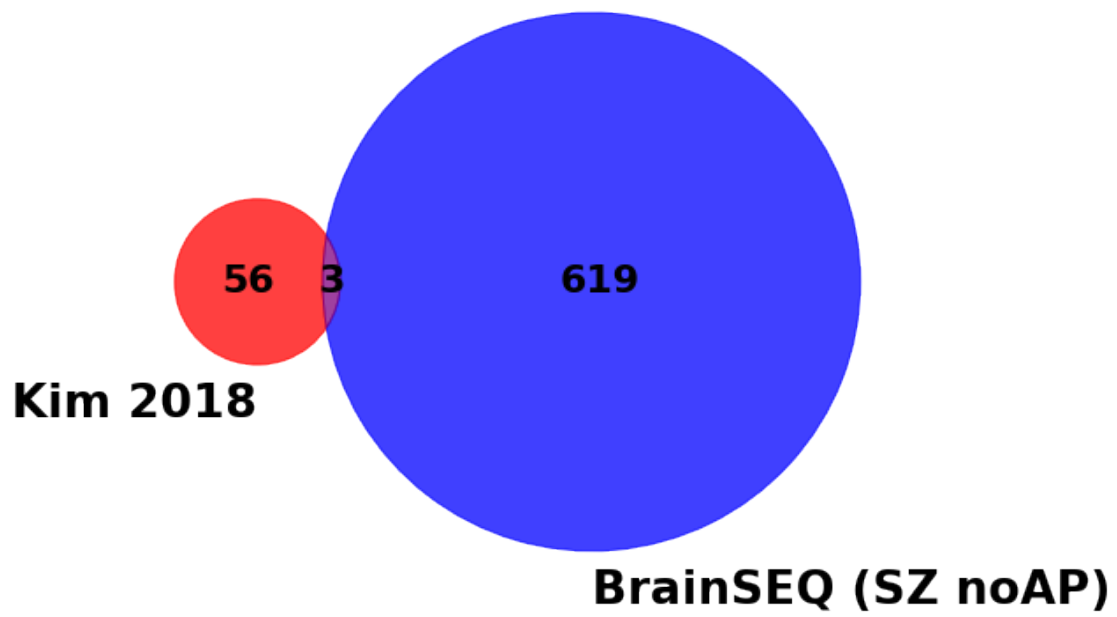




SZ AP vs control

```
[17]: plot_pairwise_venn(set(chong_2002.loc[:, "Gene Name (Human)"]), noap, "Chong_2002", "SZ noAP")
      plot_pairwise_venn(set(korostynski_2013.Symbol_human), noap, "Korostynski_2013", "SZ noAP")
      plot_pairwise_venn(set(kim_2018.Symbol_human), noap, "Kim 2018", "SZ noAP")
```



[]: