Nonpara clustering

2023-10-19

Table of contents

Simulation

- $#SNPs = $ `python print(nSNPs)` < 2000 is typical
- $#Subjects = $ `python print(nsubjects)`

```
nSNPs = 1000
nsubjects = 500
nclusts = 1
nphenos = 2
shared = 0.5
prop_causal = [0.25, 0.25]
theta_alleles = [0.75, 0.25]
h2Hom =0.5
# Add this for transfer learning
h2Het = [0, 0]

rng = np.random.default_rng()
nclusts = 2

sim = pheno_simulator(nsubjects = nsubjects, nSNPs = nSNPs)
```

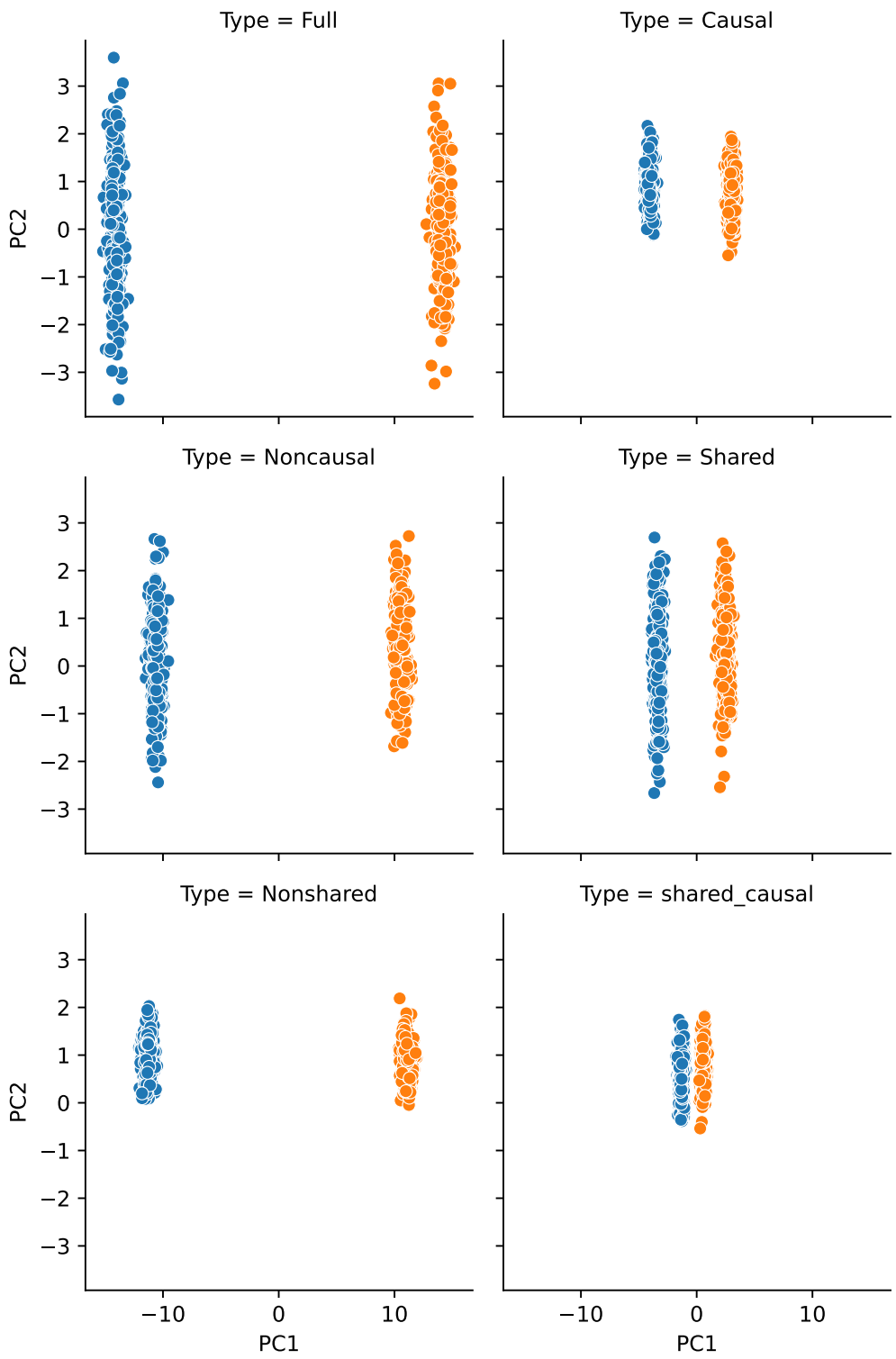
```
sim.sim_sites()
sim.sim_pops(nclusts = nclusts, theta_alleles = theta_alleles, shared = shared)
sim.sim_genos()
sim.sim_pheno(h2Hom =h2Hom, h2Het = h2Het, nphenos = nphenos, prop_causal = prop_causal, alp
# Subtract the mean from every column starting with Y and store them in the dataframe
```

```
/home/christian/Research/Stat_gen/tools/MASH/Simulate/simulation_helpers/pheno.py:79: Runtime
  prop = (cluster_freq * (1- cluster_freq)) ** alpha
/home/christian/Research/Stat_gen/tools/MASH/Simulate/simulation_helpers/pheno.py:79: Runtime
  prop = (cluster_freq * (1- cluster_freq)) ** alpha
```

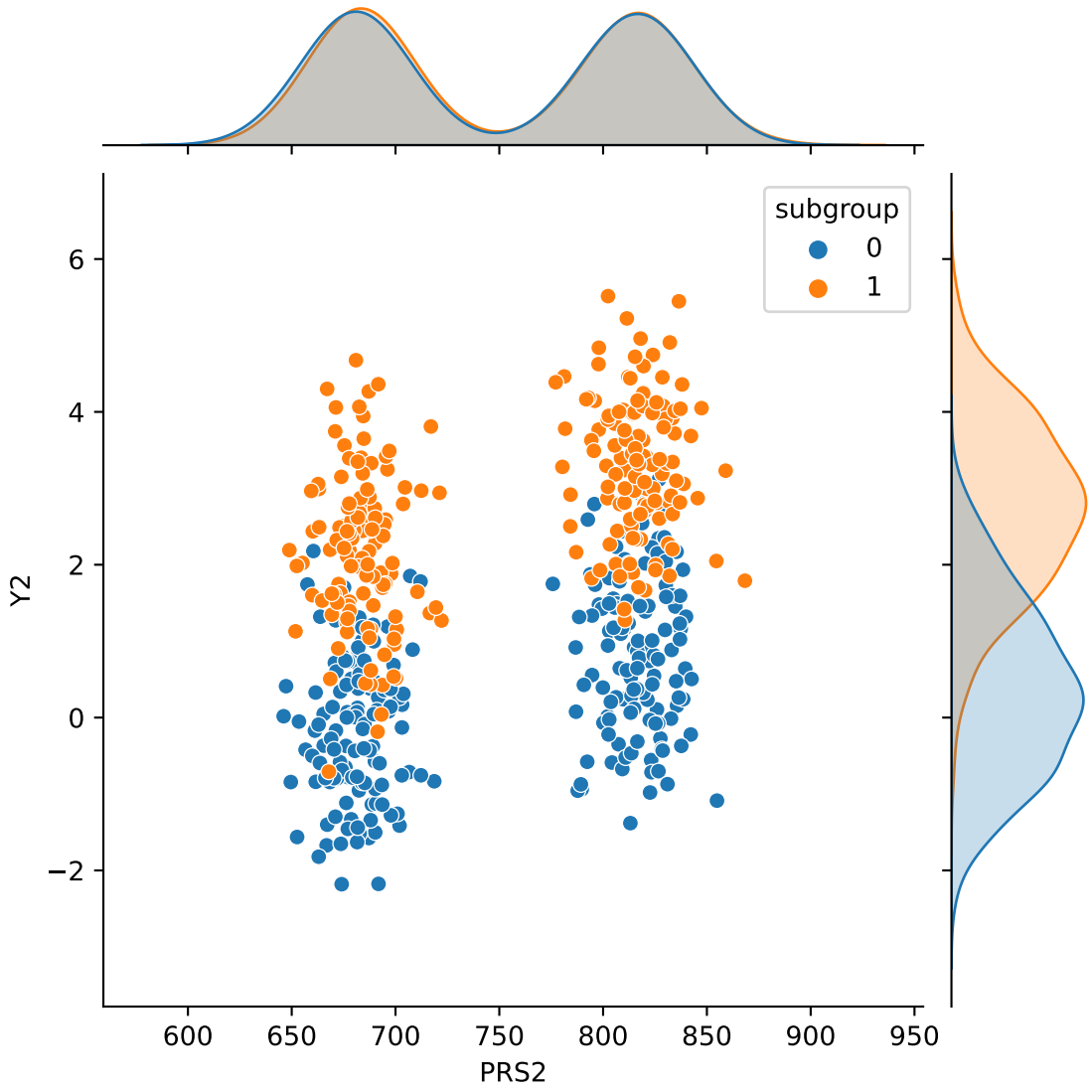
```
sim.df.iloc[:,30:(30+nphenos)] = sim.df.filter(regex='^Y') - sim.df.filter(regex= "^Y").mean

# Add a mean for subgroups
sim.df["subgroup"] = np.repeat([0, 1], 250)
sim.df.Y1 = sim.df.Y1 + sim.df.subgroup  * 2
# Create confounding variable that is related to risk subgorup
sim.df["confounder"] = np.array([rng.choice([1.0, 0.0], p = [0.75, 0.25] ) if (group == 1) el
# And affects the outcome
sim.df["Y2"] = sim.df.Y1 + sim.df.confounder
sim.df["Y3"] = sim.df.Y1 + sim.df.confounder * 3 + sim.df.subgroup * 4


# for each columns starting with letter Y, run a simple linear regression with the
GWASBetas = np.zeros(shape = (sim.genotypes.shape[1], sim.df.filter(regex="^Y").shape[1]))
for j in range(sim.df.filter(regex="^Y").shape[1]):
    for i in range(sim.genotypes.shape[1]):
        mod = OLS(sim.df["Y" + str(j)], sim.genotypes[:,i]).fit()
        if mod.pvalues[0] < 1e-2 :
          GWASBetas[i,j] = OLS(sim.df["Y" + str(j)] , sim.genotypes[:,i]).fit().params[0]
        else :
          GWASBetas[i,j] = 0
    sim.df[f"PRS{j}"] = np.matmul(sim.genotypes, GWASBetas[:,j])

np.save("genos2Anc.npy", sim.genotypes)

# Save GWAS results
np.save("gwas2Anc.npy", GWASBetas)

# Save dataframe
sim.df.to_csv("data2Anc.csv", index = False)
```

```
# SNP_props = pd.DataFrame({"Shared" : [0, 0, 1, 1],
#               "Causal" : [0, 1, 0, 1],
#               "N" : np.array([(1 - shared) * (1 - prop_causal[0]),
#                               (1 - shared) * prop_causal[0],
#                               shared * (1 - prop_causal[1]),
#                               shared * (prop_causal[1])]) * nSNPs})
# ax = sns.barplot(data= SNP_props, x = "Shared", y = "N", hue = "Causal", )
# ax.set(yscale = "log") # plotClusters(sim)
# plt.show()
```

```
plotClusters(sim)
```


Type = Full
Type = Causal
Type = Noncausal
Type = Shared
Type = Nonshared
Type = shared_causal
PC2
PC1
3
2
1
0
−1
−2
−3
−10
0
10

```
# make two subplots
# one for phenotype y0 and another for phenotype y1
p1 = sns.jointplot(x= "PRS0" , y ="Y0", data = sim.df, hue = "subgroup")
p2 = sns.jointplot(x= "PRS1" , y ="Y1", data = sim.df, hue = "subgroup")
p3 = sns.jointplot(x= "PRS2" , y ="Y2", data = sim.df, hue = "subgroup")

plt.show()
```



Data Generating mechanism

Trying to detect risk subgroups while simultatenously compute a PRS.

$$Y = X_c\beta_c + X_G\beta_G + X_R\beta_R + \epsilon$$

Implies

$$Y = X_c\beta_c + PRS\beta_{PRS} + PC\beta_{PC} + \epsilon$$

Where X is the confounding variable, X_G is the genetic variable, β is the effect of the confounding variable, β_G is the effect of the genetic variable, β_{PRS} is the effect of the PRS, β_{PC} is the effect of the principal components, X_R is the risk group, β_R is the risk group effect, and ϵ is the error term. Additionally, X_c depends on X_R

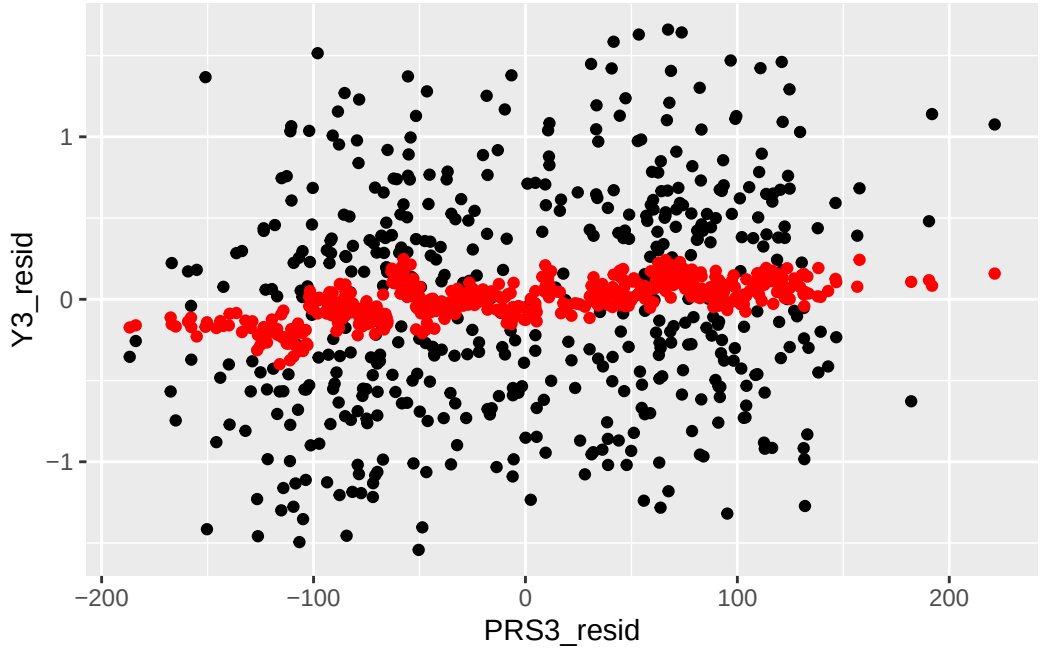
$$p(X_c|X_R) = \begin{cases} 0.75 & \text{if } X_R = 1 \\ 0.25 & \text{if } X_R = 0 \end{cases}$$

PPMx fits

Two ancestries

```
meanModel = 1
M=1e-35
similarity_function = 1
draws = 2000
burn = 500
thin = 10

# Load datafraem
df <- read_csv("data2Anc.csv")  %>%
  mutate(Y0 = (Y0 - mean(Y0))/ sd(Y0),
         Y1 = (Y1 - mean(Y1)) / sd(Y1),
         Y2 = (Y2 - mean(Y2)) / sd(Y2),
         Y3 = (Y3 - mean(Y3)) / sd(Y3),
         PRS0 = (PRS0 - mean(PRS0)) / sd(PRS0),
         PRS1 = (PRS1 - mean(PRS1)) / sd(PRS1),
         pc_1 = (pc_1 - mean(pc_1)) / sd(pc_1),
```

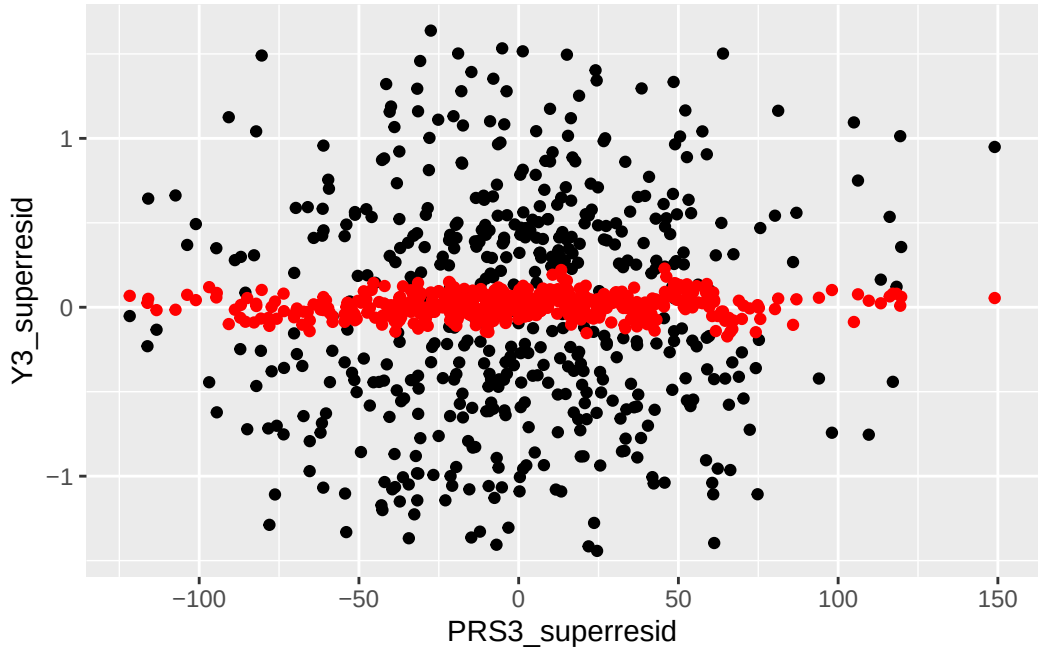
```
          )


df["pc1_resid"]<- lm(pc_1 ~ confounder, data= df)$resid
df["PRS3_resid"]<- lm(PRS3 ~ confounder, data= df)$resid
df["Y3_resid"]<- lm(Y3 ~ confounder, data= df)$resid
df["Y3_superresid"]<- lm(Y3 ~ confounder + subj_ancestries, data= df)$resid
df["PRS3_superresid"]<- lm(PRS3 ~ confounder + subj_ancestries, data= df)$resid

resid = gaussian_ppmx(y = df$Y3_resid, X = df[c("PRS3_resid", "pc1_resid")], meanModel = mean
ppmxsummary(resid, df, x ="PRS3_resid", y= "Y3_resid")
```




```
, , subj_ancestries = 0

     subgroup
label  0  1
   1   2  0
   2   2  0
   3   7 20
   4  26 18
   5   0  0
   6   0  0
```

```
   7  20 24
   8   0  2
   9  24 18
   10  5  6
   11  0  0
   12 10  7
   13  1  1
   14 12  9
   15  0  4
   16  4  4
   17  6  7
   18  0  0
   19  3  5
   20  3  0

, , subj_ancestries = 1

     subgroup
label  0  1
   1   4  4
   2   4  1
   3   0  0
   4   0  0
   5  22 27
   6  27 23
   7   0  0
   8   4  5
   9   0  0
   10  0  1
   11 26 24
   12  0  0
   13 15 16
   14  1  1
   15 14 12
   16  1  2
   17  0  2
   18  1  5
   19  5  1
   20  1  1
```

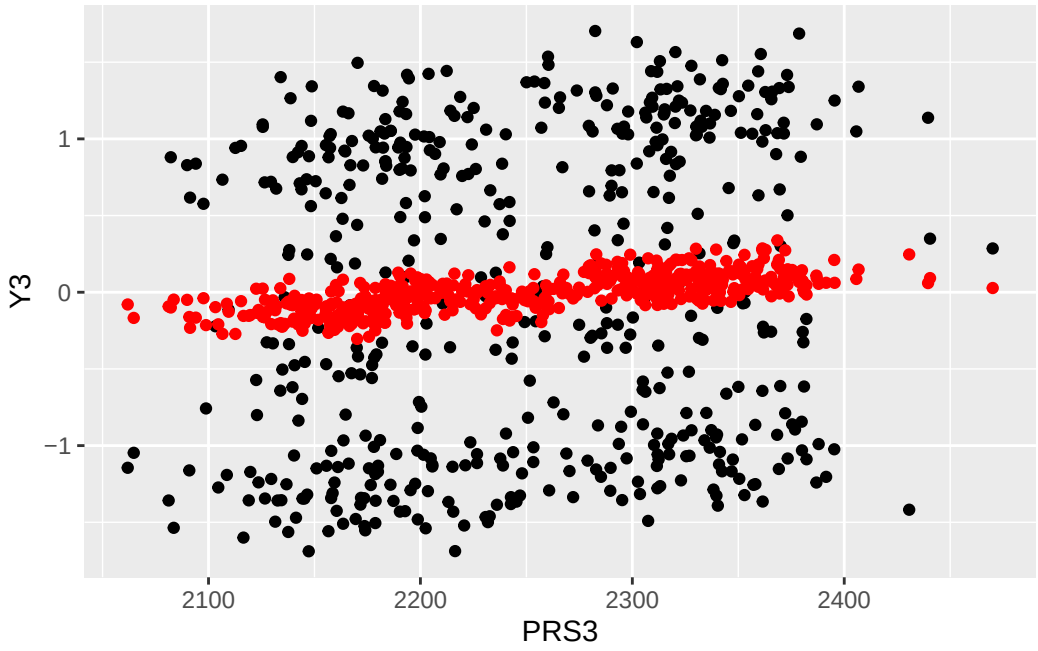
```
superresid = gaussian_ppmx(y = df$Y3_superresid, X = df["PRS3_superresid"], meanModel = meanl
ppmxsummary(superresid, df, x ="PRS3_superresid", y= "Y3_superresid")
```




```
, , subj_ancestries = 0

     subgroup
label  0  1
   1  12 13
   2   5 12
   3  10  9
   4   5  5
   5   8  9
   6   8 10
   7   7  7
   8   8  5
   9   4  1
   10  8 10
   11 14  9
   12  3  5
   13  6 10
   14  8 11
   15  8  5
   16 11  4

, , subj_ancestries = 1
```

```
      subgroup
label  0  1
   1   8  8
   2   6 11
   3   9  8
   4   8  4
   5  10 14
   6   8  6
   7   9  4
   8  10  9
   9   1  4
   10 11 10
   11  6  7
   12  1  3
   13 12  7
   14 10  6
   15  9  9
   16  7 15
```

```
ppmx = gaussian_ppmx(y = df$Y3, X = df[c("PRS3", "pc_1", "confounder")], meanModel = meanMode
ppmxsummary(ppmx, df, x ="PRS3", y= "Y3")
```




```
, , subj_ancestries = 0
```

```
     subgroup
label   0   1
    1  44  53
    2   1   5
    3  80  67

, , subj_ancestries = 1

     subgroup
label   0   1
    1  19  18
    2 106 107
    3   0   0
```