Supplementary file, S1 – elaboration on the data retrieval and statistical procedures executed.

The data that have been used for the analysis presented in *Molecular pathway analysis associates alterations in obesity related genes and antipsychotic-induced weight gain.* By H. T. Corfitsen, B. Krantz, A. Larsen, A. Drago., originates form the CATIE schizophrenia trial, that is part of the Clinical Antipsychotic Trials of Intervention Effectiveness (CATIE) Project NIMH CATIE sample (NIMH contract NO1 MH90001). Made available for analysis from https://www.nimhgenetics.org/request-access/how-to-request-access to MD PhD Antonio Drago.

Data extracted from the trial were 495.172 SNPs were available from 765 (556 males) individuals including patient descriptive data (see table S1). The data was processed through different methods, involving statistical, genetic and molecular pathway analysis. The software tools (languages) used to treat the data were R, Bash, Plink, Impute and gtool.

Table S1

Age
Age at presentation of diagnosis
Education choice
Education duration
Employment
Gender
Genetic analysis
Marital status
Maximum weight change during treatment
Race
Treatment years

Table legend: Variables available on each individual to be included in the analysis as described below

A model script is described beloved – note that the script has been colorized as to what language was used, Black == R language, Purple == Bash language / plink / impute / gtool. Where it was deemed necessary, each script line, has a comment in blue writing, as to improve the understanding of each step. This is marked with a # as it is not part of the script itself.

The following shows the first part of the analysis: upload of the database into an R environment, covariate analysis, preparation of the files for the genetic analysis; identifying the groups compared in the study i.e. patients with a record of at least one excessive weight gain period vs patients who did not gain excessive weight at any time during the study.

```
setwd("/home/[...]/CATIE_weight")
#^ this sets the directory for the analyses

#############

# a function is loaded in to the environment to facilitate the descriptive analyses (further calculations and commands are basic and not related to the data-analysis, not commented)

library(foreach)
options(warn=-1)

mx <- function (h) {

ax<-round(mean(h,na.rm=T),2)
sdx<-round(sd(h,na.rm=T),2)
result<-paste(ax,"+/-",sdx,sep="")
result
}

tabx <- function (h) {

namesx<-names(table(h))
nx<-foreach(i=1:length(names(table(h))),.combine=c) %do% table(h)[[i]][1]
nxp<-round(100*nx/sum(nx),2)
result<-paste("n=",nx, "(", nxp, "%", ")",sep="")
d<-data.frame(namesx,result)
colnames(d)<-c("Variable","n(%)")
d
}

description<-function(h) {

a<- foreach(i=1:ncol(h)) %do% {

if(class(h[,i])=="integer" | class(h[,i])=="numeric")
mx(h[,i])
else
tabx(h[,i])

}
names(a)<-colnames(h)
```

```
a

}

#############

demo<-read.csv("DEMO.csv",header=T)
     #^ this command reads the database containing the demographics of the database
vital<-read.csv("VITAL.csv")
      #^ this command reads the database containing the variables of the database containing the outcome of interest
id<-unique(vital$NIMHID)
   #^ a vector is created containing single ids, corresponding the genetic and clinical ids
library(foreach)
#^ the foreach library is uploaded in the environment, allowing for the creation of loops
weight<-foreach(i=1:length(id),.combine=c)%do%{
       #^ a loop is started
            #^ the loop with run as many times as the number of subjects in the database
a<-subset(vital, vital$NIMHID==id[i])
 #^ a database for every single subject is created in a single loop as an output of a single reiteration.
max(a$P_GAIN, na.rm=T)
#^ the max Phase Specific Percent weight Gain is recorded as the outcome of choice and put in the single database as an output of the loop
}
Weight<-data.frame(id,weight)
      #^ a dataframe containing the ids and the outcome of the loop, containing the max phase specific percent weight gain is created
colnames(Weight)<-c("NIMHID","weight")
#^ colnames to the database "Weight" are given
d1<-merge(Weight,demo,by="NIMHID")
    #^ the databases containing the demographic variables and the outcome variable are merged
stele_id<-read.delim("stele", header=F, sep = " ")
       #^ the file containing the ids as they are recorded in the plink file from the NIMH are uploaded
colnames(stele_id)<-c("cell_id","NIMHID")
             #^ the variables in the file containing the ids as in the fam file from the plink NIMH files are names, the file contains both the ids from the genetic and from the clinical file, prviding a bridge for the two datasets, a "Roseta stele".
stele_snp<-read.delim("steleSNPS", header=F)
colnames(Weight)<-c("NIMHID","weight")

d2<-merge(stele_id,d1)
   #^ a database containing both the genetic and the clinical ids is created, it will allow for the creation of a phenotype plink file for the genetic analyses.
```

```
library(car)
#^ the car libray is uploaded in the environment
d2$sex<-recode(d2$GENDER,"'Female'=0;'Male'=1")
d2$weight2<-recode(d2$weight,"-Inf=0")
#^ variables are recoded to meet the plink format
d3<-d2[c("cell_id","sex","weight2")]
#^ a new database is created containing only the variables for the plink outcome file
colnames(d3)<-c("id","sex","weight")
#^ the names of the variables of the new database are given
write.table(d3, "CATIE_weight_pheno", col.names=T, row.names=F, quote=F)
#^ the file for the plink output is prepared
t.test(weigth2,GENDER,data=d2); cor.test(d2$weight2,d2$AGE); summary(aov(weight2~MARITAL,
data=d2)); cor.test(d2$weight2,d2$EDUC_YRS); summary(aov(weight2~AT_EDUC,data=d2)); cor.te
st(d2$weight2,d2$YRS_TRT); cor.test(d2$weight2,d2$YRS_PRES); summary(aov(weight2~EMPLOY,
data=d2)):
#^ begins the study of covariates
d4<-d2[c("cell_id","sex","YRS_TRT","AGE","YRS_PRES")]
#^ a database with the covariates significantly associated with the outcome is prepared
colnames(d3)<-c("id","sex","yers_treat","age","yers_pres")
#^ the columns of the database are named
write.table(d3, "CATIE_weight_cov", col.names=T, row.names=F, quote=F)
#^ the covariate file for the plink analysis is prepared
```

The second part of the analysis follows the standard plink commands:

```
plink --noweb --bfile $name.file --covar $name.file --hwe 0.0001 --maf 0.01 –geno 0.95
```

Before imputation the pruning fase was conducted according to the parameters --indep 50 5 2 in plink environment

IMPUTATION PHASE – language is Bash, not R - *

```
# a project to impute a fam bed bim
# will use gtool and IMPUTE2
# require that the "map" file (they are not the map file from plink, though the extension is the
same in instructions) are present in the directory
# can be downloaded from https://mathgen.stats.ox.ac.uk/impute/impute_v2.html#reference
# liftOver files can be downloaded from http://hgdownload.soe.ucsc.edu/downloads.html
# the txt files that are downloaded and unpacked from the site are used as map files
# they represent the source of imputation

location="[...]For_imputation/1000GP_Phase3"
location2="[...]For_imputation"
echo -n "name of database: "
read data
```

```
# here the plink bed is turned into uscb.bed, which is the format that can be read by liftOver
plink2 --recode vcf --bfile $data --out $data
grep -v '^#' $data.vcf | awk -F '\t' '{print $1,$2,$2,$3}' > $data.ucsc.bed

echo -n "enter the name of the chain for liftOver: "
read chain

[...]liftOver $data.ucsc.bed $location2/$chain output.bed unlifted.bed



for i in {1..22}
do
plink --bfile $data --noweb --chr $i --recode --out $data.$i --maf 0.01 --geno 0.9 --hwe 0.00001
done


for i in {1..22}
do
gtool -P --ped $data.$i.ped --map $data.$i.map --og $data.$i.gen --os $data.$i.sample
done


for i in {1..22}
do
awk '{print $1}' $location/genetic_map_chr$i\_combined_b37.txt | split -l 600 -d -a4
ls x* | while read j
do
a=`head -1 $j`
b=`tail -1 $j`
impute2 \
-prephase_g -m $location/genetic_map_chr$i\_combined_b37.txt \
-g $data.$i.gen \
-int $a $b \
-Ne 20000 -o $data.$i.$j.pre-phased.impute2

impute2 \
-use_prephased_g \
-m $location/genetic_map_chr$i\_combined_b37.txt \
-int $a $b \
-Ne 20000 \
-known_haps_g $data.$i.$j.pre-phased.impute2 \
-o $data.$i.$j.phased.impute2
done
```

rm x*

done

The following shows the third part of the analysis and the core of the molecular pathway analysis along with permutation analysis.

library(ReactomePA)

^ This package provides functions for pathway analysis based on REACTOME pathway database. It implements enrichment analysis, gene set enrichment analysis and several functions for visualization

library(org.Hs.eg.db)

^ This command installs genome wide annotation for Human, primarily based on mapping using Entrez Gene identifiers

library(foreach)

^ This command installs the necessary commands to run the loop in R

library(doParallel)

^ This command installs the necessary instructions to exploit the multi-processor cores

no_cores <- detectCores() - 1

^ the number of cores to be used for the analyses are set

registerDoParallel(cores=no_cores)

^ the number of cores for be used for the analyses are registered

BiocInstaller::biocLite('grimbough/biomaRt')

library(biomaRt)

^ This command installs biomaRt, which provides an interface to a growing collection of databases implementing the BioMart software suite. It will allow to acquire information about the SNPs and genes and molecular cascades from international databases.

ensembl_gene=useMart("ENSEMBL_MART_ENSEMBL",dataset="hsapiens_gene_ensembl")

^ This command instructs the acquiring of information about genes.

ensembl_snp=useMart("ENSEMBL_MART_SNP",dataset="hsapiens_snp")

^ This command instructs the acquiring of information about SNPs.

#ensembl_fgene=useMart("ENSEMBL_MART_FUNCGEN",dataset="hsapiens_annotated_feature")

^ This command was not used in the core analysis.

load("simulation.RData")

^ This command loads previously saved data, containing SNPs significantly associated with the phenotype under analysis and SNPs of reference. "d" is the common name for "database".

d3<-steleSNP[steleSNP[,1]%in%d$SNP,]

d4<-d3[!duplicated(d3[,2]),]

```
# ^ In these lines the SNPs found to be significantly associated with the phenotype and present in the interrogated database are selected and gathered in "d4"

A<-tryCatch({foreach(i=1:100000,.combine=rbind) %dopar% {

# ^ "tryCatch" allows for the loop not to exit in even of "no match" after interrogation of the international database

#^ "foreach" starts the 2000000 loop

#^ "rbind" will combine the loop result in a database

a<-as.data.frame(head(sample(as.character(d4[,2])),1000))

#^ 1000 SNPs from d4 are shuffled

#library(biomaRt);library(ReactomePA);library(org.Hs.eg.db)

geneid<-tryCatch({getBM(attributes=c("refsnp_id","ensembl_gene_stable_id"),
filters="snp_filter",values=a[,1],mart=ensembl_snp)}, error = function(e) {return(NA)})

#^ the genes corresponding to these 1000 SNPs are identified from the ensembl_snp database

geneid2<-geneid[!duplicated(geneid[,2]),][geneid[,2]!="",]

#^ the duplicated genes are deleted

geneid3<-
tryCatch({getBM(attribute="entrezgene",filters="ensembl_gene_id",values=geneid2[,2],mart=ensembl_gene)}, error = function(e) {return(NA)})

#^ the genes are coded from the ensembl_gene database

geneid4<-geneid3[!duplicated(geneid3[,1]),]

#^ the duplicated genes are deleted

x<-enrichPathway(gene=names(table(geneid4)),pvalueCutoff=0.05, readable=T)

#^ the enrichment pathway analysis is run step 1, it accepts two inputs: the names of the genes "geneid4" and the pvalue cutoff.

as.data.frame(as.data.frame(x)[1,2:7])->b

b

#^ the enrichment pathway analysis is run step 2, this is run only to have a writable representation of the outcome, to create a database as a pruduct of the loop.

write.table(b,"A",col.names=T,row.names=F,quote=F,sep="\t",append=T)

#^ the enrichment pathway analysis is run step 3, this is run only to have a usuable representation of the outcome


} }, error = function(e) {return(NA)})

#^ loop is closed

save.image("simulation20.RData")

#^ data is saved
```