

# Statistical power

Federica Grosso 904701

2024-08-06

## Power analysis

```
library(ggplot2)
library(tidyverse)
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats    1.0.0      ✓ stringr    1.5.1
## ✓ lubridate  1.9.3      ✓ tibble     3.2.1
## ✓ purrr      1.0.2      ✓ tidyr      1.3.1
## — Conflicts — tidyverse_conflicts() —
## X dplyr::filter() masks stats::filter()
## X dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
library(ggrepel)
```

## Functions

```
explained_variance <- function(data, N)
{
  eaf = data$eaf.exposure
  MAF <- ifelse(eaf <= 0.5, eaf, 1-eaf)
  beta = data$beta.exposure
  se =data$se.exposure
  R2 = 2 * beta^2 * MAF * (1 - MAF) / (2 * beta^2 * MAF * (1 - MAF) + se^2 * 2 * N * MAF * (1 -
  MAF))
  R_2=sum(R2)
  return(R_2)
}

calculate_power_B <- function(b1, n, ratio, sig, rsq) {
  power <- pnorm(sqrt(n*rsq*(ratio/(1+ratio))*(1/(1+ratio)))*b1-qnorm(1-sig/2))*100
  return(power)
}

calculate_power_C <- function(b1, n, sig, rsq){
  power<-pnorm(sqrt(rsq*b1^2*n)-qnorm(1-sig/2))*100
  return(power)
}
```

## Power of MR analysis

# Load of IV

```
path <- "C:\\Users\\feder_phxiw9d\\Desktop\\Tesi\\Materiale_Serena\\Results\\Results_MacularD
eg\\Harm_data_E_GCST90027663_O_MACDEG.txt"
path1 <- "C:\\Users\\feder_phxiw9d\\Desktop\\Tesi\\Materiale_Serena\\Results\\Results_AMD2\\H
arm_data_E_GCST90027663_O_AMD2.txt"
path2 <- "C:\\Users\\feder_phxiw9d\\Desktop\\Tesi\\Materiale_Serena\\Results\\Results_EarlyAM
D\\Harm_data_E_GCST90027663_O_EarlyAMD.txt"
path3 <- "C:\\Users\\feder_phxiw9d\\Desktop\\Tesi\\Materiale_Serena\\Results\\Other_analysis
\\Results_Fingen\\Harm_data_E_GCST90027663_O_AMD.txt"
data <- read.table(path, sep = "\\t")
data <- data[data$mr_keep != FALSE, ]
data1 <- read.table(path1, sep = "\\t")
data1 <- data1[data1$mr_keep != FALSE, ]
data2 <- read.table(path2, sep = "\\t")
data2 <- data2[data2$mr_keep != FALSE, ]
data3 <- read.table(path3, sep = "\\t")
data3 <- data3[data3$mr_keep != FALSE, ]
```

## Calculation of explained variance $R^2$

```
N=7738
R_2 <- explained_variance(data, N)
R_21 <- explained_variance(data1, N)
R_22 <- explained_variance(data2, N)
R_23 <- explained_variance(data3, N)
# 0.01859125 con GCST90027737
# 0.01858982 con GCST90027663
R_2
```

```
## [1] 0.01858982
```

```
R_21
```

```
## [1] 0.01858982
```

```
R_22
```

```
## [1] 0.01858982
```

```
R_23
```

```
## [1] 0.01858982
```

# Calculation of power

```
n=456348  
n1=56637  
n2=105248  
n3=430221
```

```
pwr <- calculate_power_B(b1=0.2, n=456348, ratio=1295/ 455053, sig= 0.05, rsq= R_2);pwr
```

```
## [1] 16.35293
```

```
pwr <- calculate_power_B(b1=0.2, n1, ratio=3685/52952, sig= 0.05, rsq= R_2);pwr
```

```
## [1] 35.96539
```

```
pwr <- calculate_power_B(b1=0.2, n2, ratio=14034/91214, sig= 0.05, rsq= R_2);pwr
```

```
## [1] 85.25363
```

```
pwr <- calculate_power_B(b1=0.2, n3, ratio=11023/419198, sig= 0.05, rsq= R_2);pwr
```

```
## [1] 80.67811
```

## Binary outcome

```
cat("Power of analysis with", n, "participants: ", pwr, "%")
```

```
## Power of analysis with 456348 participants: 80.67811 %
```

## Plot of effect size vs power

```
# Effect size grid  
beta_values <- seq(0.05, 0.60, by=0.05)
```

```
beta_values <- seq(0.05, 0.60, by=0.05)
ratio_values <- c(1295/ 455053, 3685/52952, 14034/91214, 11023/419198)
n_values <- c(456348,56637,105248,430221)
# Inizializza un data frame vuoto per i risultati
results <- data.frame()

# Itera su ciascun valore di beta
for (beta in beta_values) {
  temp_results <- data.frame(
    Beta = beta,
    SampleSize = n_values,
    Ratio = ratio_values,
    Sigma = 0.05,
    Rsq = R_2,
    Power = mapply(calculate_power_B, beta, n_values, ratio_values, 0.05, R_2)
  )
  results <- rbind(results, temp_results)
}
```

1295/ 455053

## [1] 0.002845822

3685/52952

## [1] 0.06959133

14034/91214

## [1] 0.153858

11023/419198

## [1] 0.02629545

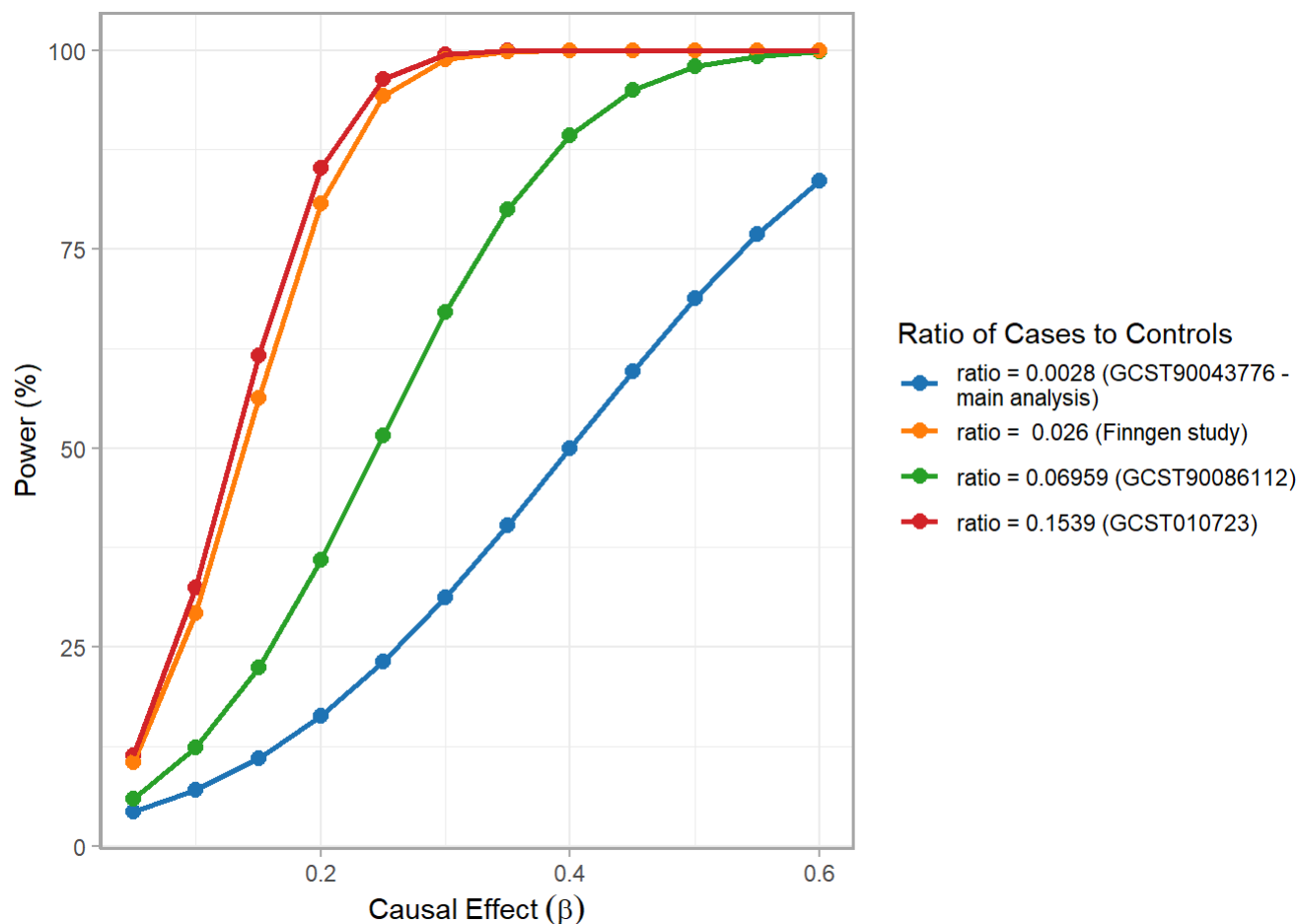
# Plot of power of AMD

```
# Filter the data to find y values where x = 0.2
annotate_data <- results %>% filter(Beta == 0.2)

ggplot(results, aes(x = Beta, y = Power, color = as.factor(Ratio))) +
  geom_point(size = 2.5) +
  geom_line(lwd = 1) +
  scale_color_manual(name = "Ratio of Cases to Controls",
    values = c("#1f77b4", "#ff7f0e", "#2ca02c", "#d62728"),
    labels = c("ratio = 0.0028 (GCST90043776 -\nmain analysis)",
      "ratio = 0.026 (Finngen study)",
      "ratio = 0.06959 (GCST90086112)",
      "ratio = 0.1539 (GCST010723)")) +
  labs(x = expression(Causal~Effect~(beta)),
    y = "Power (%)") +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "darkgrey", fill = NA, size = 1),
    axis.line = element_line(color = "darkgrey", size = 0.5),
    axis.ticks = element_line(color = "darkgrey"),
    axis.text.x = element_text(lineheight = 12),
    axis.text.y = element_text(lineheight = 12)
  )
```

```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
# geom_text(data = annotate_data,
#           aes(label = paste0(round(Power, 1), "%")),
#           size = 3.5, color = "black")
```

## LACT vs In15

### Load of IV

```
path <- "C:\\Users\\feder_phxiw9d\\Desktop\\Tesi\\Materiale_Serena\\Results\\Other_analysis
\\Results_LACT_In15\\Harm_data_E_GCST90027488_O_In15.txt"
path1 <- "C:\\Users\\feder_phxiw9d\\Desktop\\Tesi\\Materiale_Serena\\Results\\Other_analysis
\\Results_LACT_In15_REP\\Harm_data_E_GCST90027488_O_In15.txt"
path2 <- "C:\\Users\\feder_phxiw9d\\Desktop\\Tesi\\Materiale_Serena\\Results\\Other_analysis
\\Results_LACT_In15_REP\\Harm_data_E_GCST90027488_O_In15_3.txt"
data <- read.table(path, sep = "\\t")
data <- data[data$mr_keep != FALSE, ]
data1 <- read.table(path1, sep = "\\t")
data1 <- data1[data1$mr_keep != FALSE, ]
data2 <- read.table(path2, sep = "\\t")
data2 <- data2[data2$mr_keep != FALSE, ]
```

# Calculation of explained variance $R^2$

```
N=7738
R_2 <- explained_variance(data,N);R_2
```

```
## [1] 0.02877715
```

```
R_21 <- explained_variance(data1,N);R_21
```

```
## [1] 0.02567823
```

```
R_22<- explained_variance(data2,N);R_22
```

```
## [1] 0.02541375
```

## Calculation of power

```
n = 11792 # Sample size
n1 = 5368 # Sample size
n2 = 3301 # Sample size
pwr <- calculate_power_C(0.2, n,0.05, R_2)
pwr1 <- calculate_power_C(0.2, n1,0.05, R_21)
pwr2 <- calculate_power_C(0.2, n2,0.05, R_22)
```

## Continuous outcome

```
cat("Power of analysis with ", n, "participants: ",
    pwr,"%")
```

```
## Power of analysis with 11792 participants: 95.76708 %
```

```
cat("\nPower of analysis with ", n1, "participants: ",
    pwr1,"%")
```

```
##
## Power of analysis with 5368 participants: 65.10468 %
```

```
cat("\nPower of analysis with ", n2, "participants: ",
    pwr2,"%")
```

```
##
## Power of analysis with 3301 participants: 44.90248 %
```

# Plot of power of $\ln 15$

```
beta_values <- seq(0.05, 0.60, by = 0.05)
n_values <- c(n, n1, n2)
R2_values <- c(R_2, R_21, R_22)

# Inizializza un data frame vuoto per i risultati
results <- data.frame()

# Itera su ciascun valore di beta
for (beta in beta_values) {
  # Itera su ciascuna combinazione di n_values e R2_values
  for (i in 1:length(n_values)) {
    # Calcola la potenza per la combinazione corrente di Beta, SampleSize e Rsq
    power_value <- calculate_power_C(beta, n_values[i], 0.05, R2_values[i])

    # Crea un data frame temporaneo per salvare i risultati
    temp_results <- data.frame(
      Beta = beta,
      SampleSize = n_values[i],
      Sigma = 0.05,
      Rsq = R2_values[i],
      Power = power_value
    )

    # Aggiungi i risultati temporanei al data frame principale
    results <- rbind(results, temp_results)
  }
}

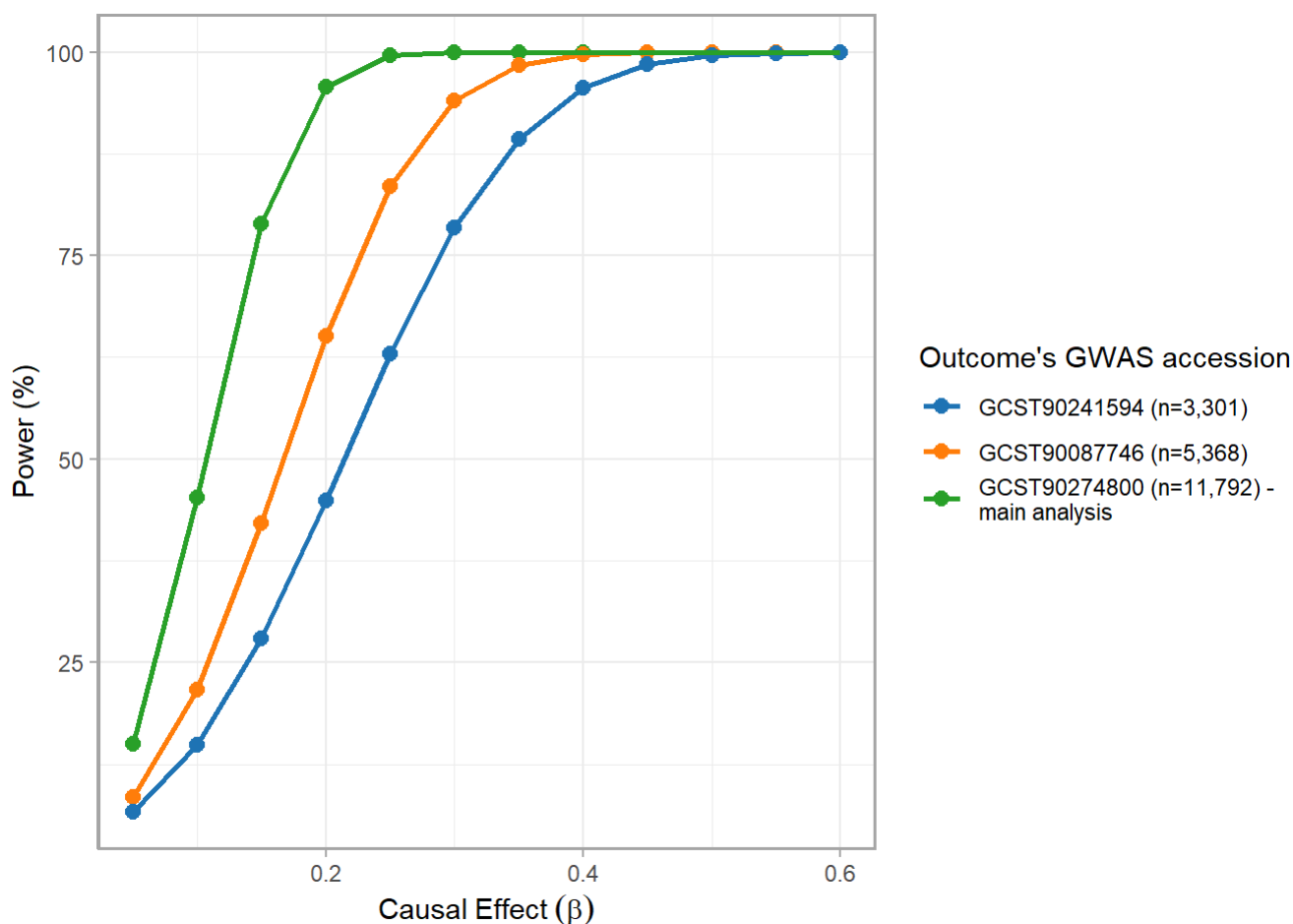
# Stampa i risultati
print(results)
```



##	Beta	SampleSize	Sigma	Rsq	Power
## 1	0.05	11792	0.05	0.02877715	14.942456
## 2	0.05	5368	0.05	0.02567823	8.488613
## 3	0.05	3301	0.05	0.02541375	6.654799
## 4	0.10	11792	0.05	0.02877715	45.309516
## 5	0.10	5368	0.05	0.02567823	21.596070
## 6	0.10	3301	0.05	0.02541375	14.823233
## 7	0.15	11792	0.05	0.02877715	78.907453
## 8	0.15	5368	0.05	0.02567823	42.117832
## 9	0.15	3301	0.05	0.02541375	27.890912
## 10	0.20	11792	0.05	0.02877715	95.767077
## 11	0.20	5368	0.05	0.02567823	65.104676
## 12	0.20	3301	0.05	0.02541375	44.902477
## 13	0.25	11792	0.05	0.02877715	99.591947
## 14	0.25	5368	0.05	0.02567823	83.526353
## 15	0.25	3301	0.05	0.02541375	62.923728
## 16	0.30	11792	0.05	0.02877715	99.981904
## 17	0.30	5368	0.05	0.02567823	94.088004
## 18	0.30	3301	0.05	0.02541375	78.459126
## 19	0.35	11792	0.05	0.02877715	99.999640
## 20	0.35	5368	0.05	0.02567823	98.419200
## 21	0.35	3301	0.05	0.02541375	89.357248
## 22	0.40	11792	0.05	0.02877715	99.999997
## 23	0.40	5368	0.05	0.02567823	99.689291
## 24	0.40	3301	0.05	0.02541375	95.578261
## 25	0.45	11792	0.05	0.02877715	100.000000
## 26	0.45	5368	0.05	0.02567823	99.955519
## 27	0.45	3301	0.05	0.02541375	98.467828
## 28	0.50	11792	0.05	0.02877715	100.000000
## 29	0.50	5368	0.05	0.02567823	99.995391
## 30	0.50	3301	0.05	0.02541375	99.559876
## 31	0.55	11792	0.05	0.02877715	100.000000
## 32	0.55	5368	0.05	0.02567823	99.999656
## 33	0.55	3301	0.05	0.02541375	99.895660
## 34	0.60	11792	0.05	0.02877715	100.000000
## 35	0.60	5368	0.05	0.02567823	99.999982
## 36	0.60	3301	0.05	0.02541375	99.979654

```
# Filter the data to find y values where x = 0.2
annotate_data <- results %>% filter(Beta == 0.2)

ggplot(results, aes(x = Beta, y = Power, color = as.factor(SampleSize))) +
  geom_point(size = 2.5) +
  geom_line(lwd = 1) +
  scale_color_manual(name = "Outcome's GWAS accession",
                    values = c("#1f77b4", "#ff7f0e", "#2ca02c"),
                    labels = c("GCST90241594 (n=3,301)",
                              "GCST90087746 (n=5,368)",
                              "GCST90274800 (n=11,792) - \nmain analysis")) +
  labs(x = expression(Causal~Effect~(beta)),
       y = "Power (%)") +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "darkgrey", fill = NA, size = 1),
    axis.line = element_line(color = "darkgrey", size = 0.5),
    axis.ticks = element_line(color = "darkgrey"),
    axis.text.x = element_text(lineheight = 12),
    axis.text.y = element_text(lineheight = 12)
  )
)
```



# LACT vs TRAIL

# Load of IV

```
path <- "C:\\Users\\feder_phxiw9d\\Desktop\\Tesi\\Materiale_Serena\\Results\\Results_IP_all
\\Harm_data_E_GCST90027488_0_GCST90274843.txt"
path1 <- "C:\\Users\\feder_phxiw9d\\Desktop\\Tesi\\Materiale_Serena\\Results\\Other_analysis
\\Results_LACT_TRAIL\\Harm_data_E_GCST90027488_0_TRAIL.txt"
data <- read.table(path, sep = "\\t")
data <- data[data$mr_keep != FALSE, ]
data1 <- read.table(path1, sep = "\\t")
data1 <- data1[data1$mr_keep != FALSE, ]
```

## Calculation of explained variance $R^2$

```
N=7738
R_2 <- explained_variance(data,N); R_2
```

```
## [1] 0.02877715
```

```
R_21 <- explained_variance(data1,N); R_21
```

```
## [1] 0.02877715
```

## Calculation of power

```
n = 14735 # Sample size
n1 = 21758 # Sample size
pwr <- calculate_power_C(0.2, n, 0.05, R_2)
pwr1 <- calculate_power_C(0.2, n1, 0.05, R_21)

cat("Power of analysis with ", n, "participants: ",
    pwr,"%")
```

```
## Power of analysis with 14735 participants: 98.45532 %
```

```
cat("\nPower of analysis with ", n1, "participants: ",
    pwr1,"%")
```

```
##
## Power of analysis with 21758 participants: 99.88349 %
```

# Plot of power of TRAIL

```
beta_values <- seq(0.05, 0.60, by = 0.05)
n_values <- c(n, n1)
R2_values <- c(R_2, R_21)

# Inizializza un data frame vuoto per i risultati
results <- data.frame()

# Itera su ciascun valore di beta
for (beta in beta_values) {
  # Itera su ciascuna combinazione di n_values e R2_values
  for (i in 1:length(n_values)) {
    # Calcola la potenza per la combinazione corrente di Beta, SampleSize e Rsq
    power_value <- calculate_power_C(beta, n_values[i], 0.05, R2_values[i])

    # Crea un data frame temporaneo per salvare i risultati
    temp_results <- data.frame(
      Beta = beta,
      SampleSize = n_values[i],
      Sigma = 0.05,
      Rsq = R2_values[i],
      Power = power_value
    )

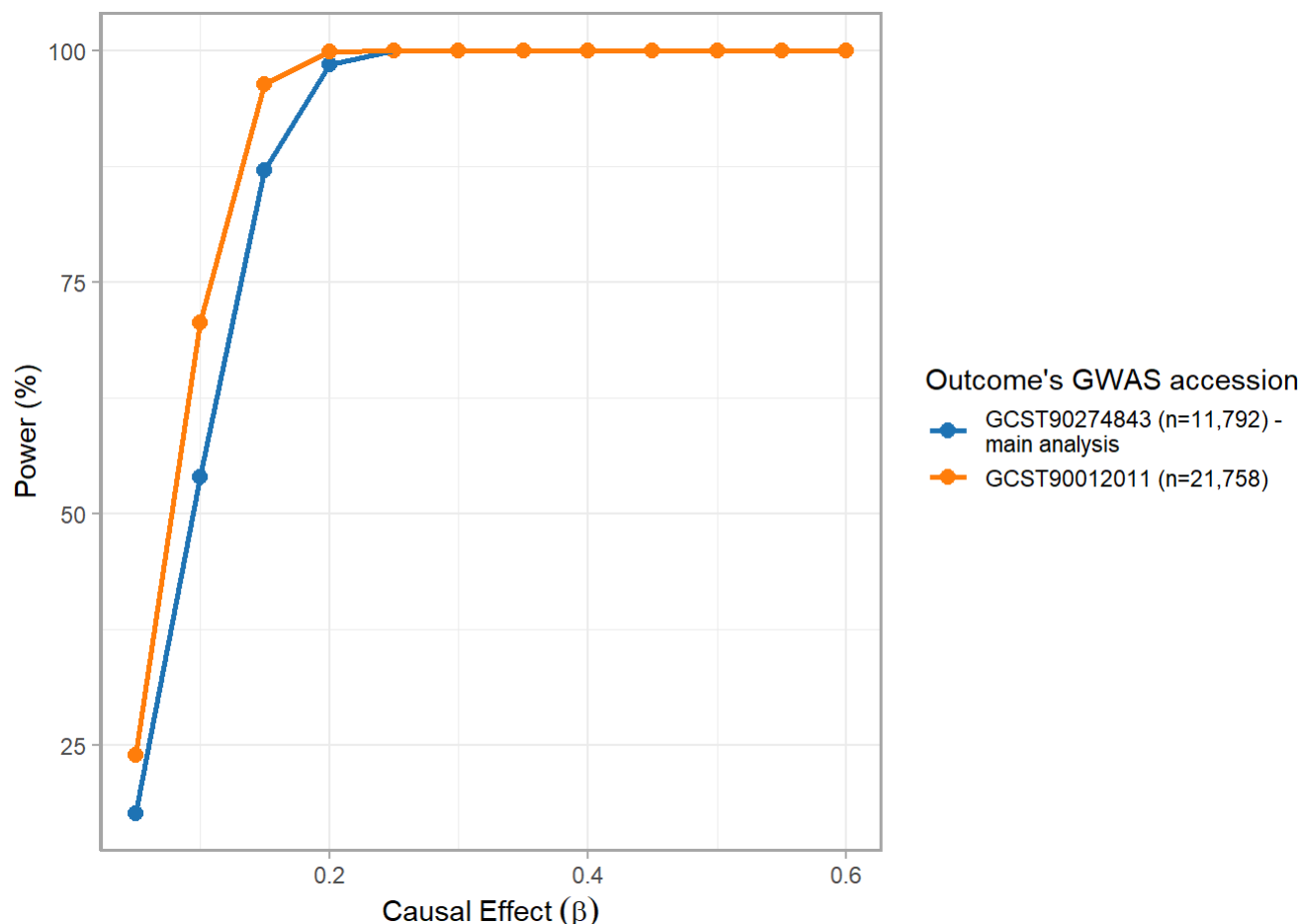
    # Aggiungi i risultati temporanei al data frame principale
    results <- rbind(results, temp_results)
  }
}

# Stampa i risultati
print(results)
```

##	Beta	SampleSize	Sigma	Rsq	Power
## 1	0.05	14735	0.05	0.02877715	17.60916
## 2	0.05	21758	0.05	0.02877715	23.92146
## 3	0.10	14735	0.05	0.02877715	53.95254
## 4	0.10	21758	0.05	0.02877715	70.61946
## 5	0.15	14735	0.05	0.02877715	87.05172
## 6	0.15	21758	0.05	0.02877715	96.35482
## 7	0.20	14735	0.05	0.02877715	98.45532
## 8	0.20	21758	0.05	0.02877715	99.88349
## 9	0.25	14735	0.05	0.02877715	99.92838
## 10	0.25	21758	0.05	0.02877715	99.99913
## 11	0.30	14735	0.05	0.02877715	99.99877
## 12	0.30	21758	0.05	0.02877715	100.00000
## 13	0.35	14735	0.05	0.02877715	99.99999
## 14	0.35	21758	0.05	0.02877715	100.00000
## 15	0.40	14735	0.05	0.02877715	100.00000
## 16	0.40	21758	0.05	0.02877715	100.00000
## 17	0.45	14735	0.05	0.02877715	100.00000
## 18	0.45	21758	0.05	0.02877715	100.00000
## 19	0.50	14735	0.05	0.02877715	100.00000
## 20	0.50	21758	0.05	0.02877715	100.00000
## 21	0.55	14735	0.05	0.02877715	100.00000
## 22	0.55	21758	0.05	0.02877715	100.00000
## 23	0.60	14735	0.05	0.02877715	100.00000
## 24	0.60	21758	0.05	0.02877715	100.00000

```
# Filter the data to find y values where x = 0.2
annotate_data <- results %>% filter(Beta == 0.2)

ggplot(results, aes(x = Beta, y = Power, color = as.factor(SampleSize))) +
  geom_point(size = 2.5) +
  geom_line(lwd = 1) +
  scale_color_manual(name = "Outcome's GWAS accession",
                     values = c("#1f77b4", "#ff7f0e"),
                     labels = c("GCST90274843 (n=11,792) - \nmain analysis",
                                "GCST90012011 (n=21,758)")) +
  labs(x = expression(Causal~Effect~(beta)),
       y = "Power (%)") +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "darkgrey", fill = NA, size = 1),
    axis.line = element_line(color = "darkgrey", size = 0.5),
    axis.ticks = element_line(color = "darkgrey"),
    axis.text.x = element_text(lineheight = 12),
    axis.text.y = element_text(lineheight = 12)
  )
```



# Bifidobacterium adolescentis vs TNFSF12

## Load IV

```
path <- "C:\\Users\\feder_phxiw9d\\Desktop\\Tesi\\Materiale_Serena\\Results\\Results_IP_all\\Harm_data_E_GCST90027754_O_GCST90274846.txt" # 1 vs 1
path1 <- "C:\\Users\\feder_phxiw9d\\Desktop\\Tesi\\Materiale_Serena\\Results\\Other_analysis\\Results_ado_TNFSF12\\Harm_data_E_GCST90032220_O_TNFSF12.txt" # 2 vs 1
path2 <- "C:\\Users\\feder_phxiw9d\\Desktop\\Tesi\\Materiale_Serena\\Results\\Other_analysis\\Results_GCST90027754_TNFSF12\\Harm_data_E_GCST90027754_O_TNFSF12.txt" # 1 vs 2
data <- read.table(path, sep = "\\t")
data <- data[data$mr_keep != FALSE, ]
data1 <- read.table(path1, sep = "\\t")
data1 <- data1[data1$mr_keep != FALSE, ]
data2 <- read.table(path2, sep = "\\t")
data2 <- data2[data2$mr_keep != FALSE, ]
```

## Calculation of explained variance $R^2$

```
N=7738
R_2 <- explained_variance(data,N); R_2
```

```
## [1] 0.02709539
```

```
N1=5959
R_21 <-explained_variance(data1,N1); R_21
```

```
## [1] 0.06742903
```

```
R_22<-explained_variance(data2,N); R_22
```

```
## [1] 0.02709539
```

## Calculation of power

```
n = 14736 # Sample size
n1 =14736 # Sample size
n2 = 3301 # Sample size
pwr <- calculate_power_C(0.2,n,0.05,R_2)
pwr1 <- calculate_power_C(0.2,n1,0.05,R_21)
pwr2 <- calculate_power_C(0.2,n2,0.05,R_22)
```

## Continuous outcome

```
cat("Power of analysis with ", n, "participants: ",
    pwr,"%")
```

```
## Power of analysis with 14736 participants: 97.9146 %
```

```
cat("\nPower of analysis with ", n1, "participants: ",
    pwr1,"%")
```

```
##
## Power of analysis with 14736 participants: 99.9993 %
```

```
cat("\nPower of analysis with ", n2, "participants: ",
    pwr2,"%")
```

```
##
## Power of analysis with 3301 participants: 47.26979 %
```

# Plot of power of TNFSF12

```
beta_values <- seq(0.05, 0.60, by = 0.05)
n_values <- c(n, n1, n2)
R2_values <- c(R_2, R_21, R_22)
ex_o <- c("e1-o1", "e2-o1", "e1-o2")
a <- 1:3

# Inizializza un data frame vuoto per i risultati
results <- data.frame()

# Itera su ciascun valore di beta
for (beta in beta_values) {
  # Itera su ciascuna combinazione di n_values e R2_values
  for (i in 1:length(n_values)) {
    # Calcola la potenza per la combinazione corrente di Beta, SampleSize e Rsq
    power_value <- calculate_power_C(beta, n_values[i], 0.05, R2_values[i])

    # Crea un data frame temporaneo per salvare i risultati
    temp_results <- data.frame(
      Beta = beta,
      SampleSize = n_values[i],
      Sigma = 0.05,
      Rsq = R2_values[i],
      Power = power_value,
      Ex_O = ex_o[i]
    )

    # Aggiungi i risultati temporanei al data frame principale
    results <- rbind(results, temp_results)
  }
}

# Stampa i risultati
print(results)
```



##	Beta	SampleSize	Sigma	Rsq	Power	Ex_O
## 1	0.05	14736	0.05	0.02709539	16.830944	e1-o1
## 2	0.05	14736	0.05	0.06742903	35.053897	e2-o1
## 3	0.05	3301	0.05	0.02709539	6.849481	e1-o2
## 4	0.10	14736	0.05	0.02709539	51.524757	e1-o1
## 5	0.10	14736	0.05	0.06742903	88.341495	e2-o1
## 6	0.10	3301	0.05	0.02709539	15.523726	e1-o2
## 7	0.15	14736	0.05	0.02709539	85.020801	e1-o1
## 8	0.15	14736	0.05	0.06742903	99.718279	e2-o1
## 9	0.15	3301	0.05	0.02709539	29.413034	e1-o2
## 10	0.20	14736	0.05	0.02709539	97.914603	e1-o1
## 11	0.20	14736	0.05	0.06742903	99.999302	e2-o1
## 12	0.20	3301	0.05	0.02709539	47.269792	e1-o2
## 13	0.25	14736	0.05	0.02709539	99.879939	e1-o1
## 14	0.25	14736	0.05	0.06742903	100.000000	e2-o1
## 15	0.25	3301	0.05	0.02709539	65.703294	e1-o2
## 16	0.30	14736	0.05	0.02709539	99.997265	e1-o1
## 17	0.30	14736	0.05	0.06742903	100.000000	e2-o1
## 18	0.30	3301	0.05	0.02709539	80.982388	e1-o2
## 19	0.35	14736	0.05	0.02709539	99.999976	e1-o1
## 20	0.35	14736	0.05	0.06742903	100.000000	e2-o1
## 21	0.35	3301	0.05	0.02709539	91.151062	e1-o2
## 22	0.40	14736	0.05	0.02709539	100.000000	e1-o1
## 23	0.40	14736	0.05	0.06742903	100.000000	e2-o1
## 24	0.40	3301	0.05	0.02709539	96.584714	e1-o2
## 25	0.45	14736	0.05	0.02709539	100.000000	e1-o1
## 26	0.45	14736	0.05	0.06742903	100.000000	e2-o1
## 27	0.45	3301	0.05	0.02709539	98.915786	e1-o2
## 28	0.50	14736	0.05	0.02709539	100.000000	e1-o1
## 29	0.50	14736	0.05	0.06742903	100.000000	e2-o1
## 30	0.50	3301	0.05	0.02709539	99.718617	e1-o2
## 31	0.55	14736	0.05	0.02709539	100.000000	e1-o1
## 32	0.55	14736	0.05	0.06742903	100.000000	e2-o1
## 33	0.55	3301	0.05	0.02709539	99.940568	e1-o2
## 34	0.60	14736	0.05	0.02709539	100.000000	e1-o1
## 35	0.60	14736	0.05	0.06742903	100.000000	e2-o1
## 36	0.60	3301	0.05	0.02709539	99.989818	e1-o2

```
# Filter the data to find y values where x = 0.2
annotate_data <- results %>% filter(Beta == 0.2)

ggplot(results, aes(x = Beta, y = Power, color = as.factor( Ex_0 ))) +
  geom_point(size = 2.5) +
  geom_line(lwd = 1) +
  scale_color_manual(name = "Exposure-Outcome pairs",
                    values = c("#1f77b4", "#ff7f0e", "#2ca02c"),
                    labels = c("Exposure: GCST90027754\nOutcome: GCST90243141\n(main analysis)", "Exposure: GCST90027754\nOutcome: GCST90274846", "Exposure: GCST90032220\nOutcome: GCST90243141")) +
  labs(x = expression(Causal~Effect~(beta)),
       y = "Power (%)") +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "darkgrey", fill = NA, size = 1),
    axis.line = element_line(color = "darkgrey", size = 0.5),
    axis.ticks = element_line(color = "darkgrey"),
    axis.text.x = element_text(lineheight = 12),
    axis.text.y = element_text(lineheight = 12)
  ) + theme(legend.spacing.y = unit(12, 'pt'), legend.key.size = unit(2, 'lines'))
```

