Arkesh Das CMSE 410 Semester Project

Arkesh Das

03/30/2025

Assessing the Influence of False Discovery Rate Methods on Genetic Associations in an Immune Response GWAS

Background

Genome-wide association studies (GWAS) have emerged as powerful tools for linking genetic variants with complex traits, including immune responses. A typical GWAS involves testing millions of single nucleotide polymorphisms (SNPs) for associations, which leads to a substantial multiple testing burden. Applying a simple p-value threshold (e.g., 0.05) is inadequate, as it would generate many false positives. Thus, false discovery rate (FDR) control has become an essential strategy to address this challenge.

About the data

This project utilizes data from the Milieu Intérieur (MI) project, a comprehensive, population-based study coordinated by Prof. Lluis Quintana-Murci and Dr. Darragh Duffy at Institut Pasteur in Paris. Established under the French Government's Investissement d'Avenir – Laboratoire d'Excellence (LabEx) initiative, the MI project aims to dissect the interplay between genetics, environment, and immune variation [1].

It was designed to address a critical discrepancy in medicine: while immune responses vary widely among individuals, medical care and therapeutic strategies are often standardized across populations.

I chose to use this dataset because it offers a unique opportunity to evaluate the impact of false discovery rate methods on GWAS results in the context of immune variation, and the data is high-resolution and well-characterized so it's ideal for benchmarking statistical methods.

The data was accessed from the NHGRI-EBI GWAS Catalog. The Catalog is a publicly available repository of SNP-trait associations identified in published GWAS studies. It provides standardized annotations, p-values, effect sizes, and sample information for millions of SNPs across hundreds of phenotypes.

The MI cohort consists of 1,000 healthy individuals, age- and sex-stratified across five decades (ages 20–70), and of french origin. Detailed serological data were collected, including total levels of IgA, IgE, IgG, and IgM, along with antibody responses to 15 antigens from common infectious agents and vaccines (e.g., influenza, EBV, CMV, HSV, rubella). The cohort was genotyped for over 700,000 SNPs and subsequently imputed to yield over 12 million genetic variants. Exten-

sive phenotypic, demographic, clinical, and environmental metadata are also available, including vaccine history, infection exposure, CRP levels, lipid profiles, and lifestyle factors [2].

Loading in the Data

Reading in the Data and making dataframes

```
#df will contain all of the data from the study that had MAF > 5%
#df2 is 1,000 snps, used to test code chunks before running on larger d
#df3 contains 100,000 random snps, used as an analogue for the entire d
#ch6 contains the 354095 snps identified on chromosome 6

#reloading the 'df' dataframe takes a long time because of the number o
df <- read.table("/Users/arkeshdas/Documents/CMSE 410/repos/HBV_HBc_GWAS
colnames(df) <- c("#CHROM", "POS", "ID", "REF", "ALT", "ALT_FREQ", "TEST
head(df)
```

```
##   #CHROM    POS        ID REF ALT ALT_FREQ TEST OBS_CT     OR     SE
## 1      1 729679  rs4951859   G   C 0.153770  ADD    504 1.85558 0.484720
## 2      1 731718 rs142557973   T   C 0.111111  ADD    504 1.60082 0.535842
## 3      1 734349 rs141242758   T   C 0.107143  ADD    504 1.69814 0.544585
## 4      1 736289  rs79010578   T   A 0.125000  ADD    504 1.47914 0.551837
## 5      1 752566  rs3094315   A   G 0.143849  ADD    504 2.22132 0.453686
```

```
## 6    1 752721  rs3131972  G  A 0.143849  ADD   504 2.22132 0.453686
##     T_STAT         P
## 1 1.275370 0.2021780
## 2 0.878092 0.3798940
## 3 0.972362 0.3308700
## 4 0.709382 0.4780870
## 5 1.759150 0.0785516
## 6 1.759150 0.0785516
```

```
df2 <- read.table("/Users/arkeshdas/Documents/CMSE 410/repos/HBV_HBc_GWA
colnames(df2) <- c("#CHROM", "POS", "ID", "REF", "ALT", "ALT_FREQ", "TES
head(df2)
```

```
##  #CHROM   POS        ID REF ALT ALT_FREQ TEST OBS_CT     OR     SE
## 1    1 729679  rs4951859  G  C 0.153770  ADD   504 1.85558 0.484720
## 2    1 731718 rs142557973  T  C 0.111111  ADD   504 1.60082 0.535842
## 3    1 734349 rs141242758  T  C 0.107143  ADD   504 1.69814 0.544585
## 4    1 736289  rs79010578  T  A 0.125000  ADD   504 1.47914 0.551837
## 5    1 752566  rs3094315  A  G 0.143849  ADD   504 2.22132 0.453686
## 6    1 752721  rs3131972  G  A 0.143849  ADD   504 2.22132 0.453686
##     T_STAT         P
## 1 1.275370 0.2021780
## 2 0.878092 0.3798940
## 3 0.972362 0.3308700
```

```
## 4 0.709382 0.4780870
## 5 1.759150 0.0785516
## 6 1.759150 0.0785516
```

```r
set.seed(03302025)  # Setting seed for reproducibility for df3
df3 <- df[sample(nrow(df), 100000), ]
colnames(df3) <- c("#CHROM", "POS", "ID", "REF", "ALT", "ALT_FREQ", "TES
head(df3)
```

```
##       #CHROM     POS       ID REF ALT  ALT_FREQ TEST OBS_CT     OR
## 5133272    20  58001534  rs6123895   G   A 0.0843254  ADD   504 0.515262
## 2864968     8 123959201 rs35276815   A   G 0.0496032  ADD   504 0.609257
## 2891050     8 135594129 rs72719766   G   A 0.2023810  ADD   504 1.117420
## 4067132    13  82262514  rs9531236   T   C 0.4017860  ADD   504 1.221140
## 1960339     6   3294824  rs9378357   G   T 0.0783730  ADD   504 0.846474
## 3732300    12  24216722  rs7964057   A   G 0.2896830  ADD   504 1.328880
##                SE    T_STAT        P
## 5133272 1.002900 -0.661160 0.508510
## 2864968 1.067520 -0.464174 0.642523
## 2891050 0.449246  0.247121 0.804815
## 4067132 0.415559  0.480766 0.630683
## 1960339 0.771480 -0.216047 0.828951
## 3732300 0.386534  0.735603 0.461973
```

```r
ch6 <- df %>% filter(`#CHROM` == 6)


colnames(ch6) <- c("#CHROM", "POS", "ID", "REF", "ALT", "ALT_FREQ", "TES
head(ch6)
```

```
##   #CHROM    POS        ID REF ALT  ALT_FREQ TEST OBS_CT     OR     SE
## 1      6 190248 rs115458869   A   G 0.1736110  ADD   504 1.053850 0.506894
## 2      6 191736   rs9502903   A   G 0.3988100  ADD   504 1.966050 0.392648
## 3      6 192106  rs12191783   G   A 0.1567460  ADD   504 0.417576 0.748029
## 4      6 192181  rs61430413   A   G 0.0823413  ADD   504 5.636890 0.521984
## 5      6 192243 rs115258892   C   A 0.0734127  ADD   504 6.031840 0.522290
## 6      6 192300   rs6915646   T   G 0.2242060  ADD   504 1.124090 0.441977
##       T_STAT           P
## 1  0.103479 0.917583000
## 2  1.721710 0.085121500
## 3 -1.167450 0.243027000
## 4  3.313000 0.000923018
## 5  3.440720 0.000580171
## 6  0.264669 0.791265000
```

```r
df_clean <- df %>%
  filter(!is.na(P))
```

Creating Manhattan Plots

Because I will be making many Manhattan plots, I initially decided to create a function that I can pass parameters into rather than creating a Manhattan plot by hand each time.

```
generate_manhattan_plot <- function(df,
                                    chrom_col = "#CHROM",
                                    pos_col = "POS",
                                    pval_col = "P",
                                    plot_title = "Manhattan Plot",
                                    pval_cutoff = 0.05) {

  # Step 1: Normalize CHR and BP
  df <- df %>%
    mutate(CHR = gsub("chr", "", .data[[chrom_col]], ignore.case = TRUE)
           CHR = gsub("[^0-9XYMT]", "", CHR),
           CHR = as.character(CHR),
           BP = as.numeric(.data[[pos_col]])) %>%
    filter(!is.na(BP) & !is.na(CHR))

  # Step 2: Sort
  df <- df %>% arrange(as.numeric(CHR), BP)

  # Step 3: Chromosome summary
```

```r
df_chr <- df %>%
  group_by(CHR) %>%
  summarise(max_bp = max(BP, na.rm = TRUE), .groups = "drop") %>%
  arrange(as.numeric(CHR)) %>%
  mutate(cumulative = cumsum(max_bp) - max_bp)

# Step 4: Merge
df <- df %>%
  left_join(df_chr, by = "CHR") %>%
  filter(!is.na(cumulative)) %>%
  mutate(BP_cum = BP + cumulative)

# Step 5: Axis labels
axis_df <- df %>%
  group_by(CHR) %>%
  summarise(center = (min(BP_cum) + max(BP_cum)) / 2, .groups = "drop"

# Step 6: Color map
chroms <- sort(unique(df$CHR))
chrom_colors <- rep(c("grey", "skyblue"), length.out = length(chroms))
names(chrom_colors) <- chroms
```

```
  # Step 7: Manhattan Plot
  ggplot(df, aes(x = BP_cum, y = -log10(.data[[pval_col]]))) +
    geom_point(aes(color = CHR), alpha = 0.75, size = 1.3) +
    scale_color_manual(values = chrom_colors) +
    scale_x_continuous(label = axis_df$CHR, breaks = axis_df$center) +
    labs(x = "Chromosome",
         y = expression(-log[10](p-value)),
         title = plot_title) +
    geom_hline(yintercept = -log10(pval_cutoff), linetype = "dashed", co
    theme_minimal() +
    theme(legend.position = "none",
          panel.border = element_blank(),
          panel.grid.major.x = element_blank())
}
```

```
generate_manhattan_plot(df3, pval_cutoff = 0.05, plot_title = "Manhattan
```

However, after doing some research, I learned there are already existing functions that create Manhattan plots (`qqman` library), so I will be using this function to create my Manhattan plots instead [3].

```
df3_clean <- df3 %>%
  filter(!is.na(P))
manhattan(df3_clean, chr = "#CHROM", bp = "POS", snp = "ID", p = "P", su
          genomewideline = -log10(2.7e-09), ylim = c(0, 10))
title(main = "Manhattan Plot of Raw P-values for 100,000 SNPs Bonferroni
abline(h = -log10(9.5e-09), col = "darkgreen", lty = 3)
```

```
ch6_clean <- ch6 %>%
  filter(!is.na(P))
manhattan(ch6_clean, chr = "#CHROM", bp = "POS", snp = "ID", p = "P", hi
          genomewideline = -log10(0.05), ylim = c(0, 6))
title(main = "Manhattan Plot of Raw P-values for Chr. 6 SNPs p = 0.05")
```

QQ plots

I tried to create my code to create a QQ plot before I realized that there already is one in `qqman`

```
pvals <- df3$P

pvals <- pvals[!is.na(pvals)]


n <- length(pvals)


expected <- -log10(ppoints(n))

observed <- -log10(sort(pvals))


plot(expected, observed,
```

```
     pch = 19,
     col = rgb(0, 0, 0, 0.5),
     cex = 0.6,
     xlab = "Expected -log10(p)",
     ylab = "Observed -log10(p)",
     main = "QQ Plot of GWAS P-values")

abline(0, 1, col = "red", lwd = 2)
```

```
qq(df3_clean$P, main = "QQ Plot of GWAS P-values for 100k SNPs")
```

```r
qq(ch6_clean$P, main = "QQ Plot of GWAS P-values for Chr. 6")
```

Various FDR methods

Now I will be trying different FDR methods, from most to least conservative.

Benjamini-Yekutieli

Out of the p-value adjustment methods, Benjamini Yekuteli is the most conservative, so I will try it first. R has a `p.adjust` function that can automatically apply the BY method.

```
df3_clean$P_BY <- p.adjust(df3_clean$P, method = "BY")


# Summary stats of BY-adjusted p-values
summary(df3_clean$P_BY)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1
```

```
# Filter significant results under BY
df3_by_sig <- df3_clean %>% filter(P_BY < 0.05)


manhattan(df3_clean, chr = "#CHROM", bp = "POS", snp = "ID", p = "P_BY",
title(main = "Manhattan Plot of BY adjusted P-values for 100,000 SNPs")
```

I cannot visualize the BY-adjusted p-values using my Manhattan Plot. In this case, I can thankfully modify and re-use my plotting function from earlier:

```
generate_plot <- function(df,
                                        chrom_col = "#CHROM",
                                        pos_col = "POS",
                                        pval_col = "P",
                                        plot_title = "Plot") {


  # Step 1: Normalize CHR and BP
  df <- df %>%
    mutate(CHR = gsub("chr", "", .data[[chrom_col]], ignore.case = TRUE)
```

```
          CHR = gsub("[^0-9XYMT]", "", CHR),
          CHR = as.character(CHR),
          BP = as.numeric(.data[[pos_col]])) %>%
  filter(!is.na(BP) & !is.na(CHR))

# Step 2: Sort
df <- df %>% arrange(as.numeric(CHR), BP)

# Step 3: Chromosome summary
df_chr <- df %>%
  group_by(CHR) %>%
  summarise(max_bp = max(BP, na.rm = TRUE), .groups = "drop") %>%
  arrange(as.numeric(CHR)) %>%
  mutate(cumulative = cumsum(max_bp) - max_bp)

# Step 4: Merge
df <- df %>%
  left_join(df_chr, by = "CHR") %>%
  filter(!is.na(cumulative)) %>%
  mutate(BP_cum = BP + cumulative)

# Step 5: Axis labels
```

```r
axis_df <- df %>%
  group_by(CHR) %>%
  summarise(center = (min(BP_cum) + max(BP_cum)) / 2, .groups = "drop"


# Step 6: Color map
chroms <- sort(unique(df$CHR))
chrom_colors <- rep(c("grey", "skyblue"), length.out = length(chroms))
names(chrom_colors) <- chroms


# Step 7: Plot
ggplot(df, aes(x = BP_cum, y = .data[[pval_col]])) +
  geom_point(aes(color = CHR), alpha = 0.75, size = 1.3) +
  scale_color_manual(values = chrom_colors) +
  scale_x_continuous(label = axis_df$CHR, breaks = axis_df$center) +
  ylim(0,0.007)+
  labs(x = "Chromosome",
       y = 'p-value',
       title = plot_title) +
  theme_minimal() +
  theme(legend.position = "none",
        panel.border = element_blank(),
        panel.grid.major.x = element_blank())
```

```
}
```

```
generate_plot(df3_clean, pval_col = "P_BY", plot_title = "Plot of BY adj
```

Using my new and improved plotting function, we can see that because the Benjamini-Yekutieli method is so conservative, all of the p-values were adjusted to 1.

Benjamini-Hochberg

Now I will try adjust my p-values using the Benjamini-Hochberg method.

```
# Adjusting p_values using BH

df3_clean$P_BH <- p.adjust(df3_clean$P, method = "BH")
```

```r
#Summary stats of adjusted p-values

summary(df3_clean$P_BH)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.9943  0.9943  0.9943  0.9950  0.9958  1.0000
```

```r
df3_bh_sig <- df3_clean %>% filter(P_BH < 0.05) #there are no significar

df3_clean$log10_P_BH <- -log10(df3_clean$P_BH)


manhattan(df3_clean, chr = "#CHROM", bp = "POS", snp = "ID", p = "P_BH",

title(main = "Plot of BH adjusted P-values for 100,000 SNPs")
```

```r
generate_plot(df3_clean, pval_col = "P_BH", plot_title = "Plot of BH adj
```

Benjamini-Krieger-Yekutieli

Now I will try to use the Benjamini-Krieger-Yekutieli method in the `mutoss` package [4]:

```r
mutoss::loadMethod("BKY")




bky_result <- mutoss::runMethod("BKY", list(pValues = df3_clean$P, alpha
```

Unfortunately, I could not get the `mutoss` package's implementation of BKY to work, so I will instead follow the original 2006 paper [5], which describes how to implement the BKY method:

```r
#BYK function
bky_adjust <- function(p, alpha = 0.05) {
  m <- length(p)
  p_order <- order(p)
  p_sorted <- p[p_order]
  k <- m:1
  c_m <- sum(1 / (1:m))

  thresh <- (k / m) * (alpha / (1 + alpha)) * (1 / c_m)
  test <- p_sorted <= thresh
  max_k <- if (any(test)) max(which(test)) else 0

  adjusted <- rep(1, m)
  if (max_k > 0) {
    adjusted[p_order[1:max_k]] <- pmin(1, thresh[max_k])
  }

  return(adjusted)
}

# Applying BKY to the data
```

```r
df3_clean$P_BKY <- bky_adjust(df3_clean$P, alpha = 0.05)

df3_bky_sig <- df3_clean %>% filter(P_BKY < 0.05)


summary(df3_bky_sig$P_BKY)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.003935 0.003935 0.003935 0.003935 0.003935 0.003935
```

```r
# Plot
manhattan(df3_clean, chr = "#CHROM", bp = "POS", snp = "ID", p = "P_BKY"
          suggestiveline = FALSE)
title(main = "Plot of BKY Adjusted P-values for 100,000 SNPs")
```

```
generate_plot(df3_clean, pval_col = "P_BKY", plot_title = "Plot of BKY a
```

This time, I was able to isolate around 300 significant SNPs using the BYK method. However, they were all the same p-value. Therefore, while BYK is able to separate

Storey-Tibshirani (q-value)

Now I will be adjusting my p-values using the Storey-Tibshirani method [6]. Thankfully, there is a package in R that was designed in collaboration with the creators of this method, the `qvalue` package. I followed the documentation of the `qvalue` package to use the `qvalue()` method and other methods on my data [7].

```
qobj <- qvalue(p = df3_clean$P)   # estimating pi_0 and q-values
df3_clean$q_value <- qobj$qvalues
```

```r
df3_q_sig <- df3_clean %>% filter(q_value < 0.05)

nrow(df3_q_sig)
```

```
## [1] 0
```

```r
qobj$pi0
```

```
## [1] 0.9716845
```

Unfortunately, I was not able to actually find any significant q_values. My pi_0 was 0.971, which means that 97.1% of the tested SNPs are estimated to be null.

```r
top_bky <- df3_clean %>%

  arrange(P) %>%

  slice(1:50) %>%

  select(`#CHROM`, POS, ID, P, P_BKY)
```

Taking a second look at the p-values that were marked as being signficant from BYK, they appear to be randomly clustered throughout the genome. However, the second most significant SNP is located on chromosome 6, the same chromosome as the HLA gene cluster that was found to be significant in the original study. So, I think to conclude, I will try to figure out where this SNP is located and its potential biological function.

SNP POS: 129948488 #CHROM: 6

I used the NCBI Human Genome Data Viewer [8] to figure out the genomic context of chr6:129948488. This SNP lies within an intron region of L3MBTL3, situated between a nearby HLA class II gene and an uncharacterized gene. L3MBTL3 has been implicated in several GWAS

as influencing hematopoietic traits and immunoglobulin A (IgA) levels. Variants in L3MBTL3 have been associated with natural variation in IgA concentrations, suggesting a potential role in humoral immune regulation. These findings suggest that SNPs in or near L3MBTL3 could indirectly modulate immune responses, potentially by influencing the development or regulation of immune effector cells [9].

Therefore, while I was not able to identify a clear enrichment of significant SNPs in known immune loci across the genome, I was able to pinpoint at least one biologically plausible candidate. Its proximity to the HLA region and its known association with immune-related traits make it a promising lead for further investigation. Further analysis of the SNPs that were flagged as being significant by the BKY method should be looked into, as they may represent novel loci of biological relevance that have not yet been characterized in the context of immune response by the original study.

#Misc graphs

```
df3_clean$logP <- -log10(df3_clean$P)




ggplot(df3_clean, aes(x = -log10(P))) +
  geom_histogram(bins = 50, fill = "skyblue", color = "white") +
  labs(title = "Distribution of -log10(P-values)",
       x = "-log10(P-value)", y = "Count") +xlim(0, 4)
```

```
  theme_minimal()
```

```
## List of 136
##  $ line                            :List of 6
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                            :List of 5
```

```
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                            :List of 11
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##  ..$ margin     : 'margin' num [1:4] 0points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                           : NULL
##  $ aspect.ratio                    : NULL
##  $ axis.title                      : NULL
```

```
##  $ axis.title.x                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 2.75points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
```

```
##  ..$ margin     : 'margin' num [1:4] 0points 0points 2.75points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom             : NULL
##  $ axis.title.y                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 0points 2.75points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left               : NULL
##  $ axis.title.y.right              :List of 11
##   ..$ family       : NULL
```

```
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 0points 0points 0points 2.75points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                       :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
```

```
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x                     :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 2.2points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top                 :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
```

```
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 0points 0points 2.2points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom              : NULL
##  $ axis.text.y                     :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 0points 2.2points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left                : NULL
```

```
##  $ axis.text.y.right                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 0points 0points 0points 2.2points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.theta                  : NULL
##  $ axis.text.r                      :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0.5
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
```

```
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 0points 2.2points 0points 2.2point
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                      : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.ticks.x                    : NULL
##  $ axis.ticks.x.top                : NULL
##  $ axis.ticks.x.bottom             : NULL
##  $ axis.ticks.y                    : NULL
##  $ axis.ticks.y.left               : NULL
##  $ axis.ticks.y.right              : NULL
##  $ axis.ticks.theta                : NULL
##  $ axis.ticks.r                    : NULL
##  $ axis.minor.ticks.x.top          : NULL
##  $ axis.minor.ticks.x.bottom       : NULL
##  $ axis.minor.ticks.y.left         : NULL
##  $ axis.minor.ticks.y.right        : NULL
##  $ axis.minor.ticks.theta          : NULL
##  $ axis.minor.ticks.r              : NULL
##  $ axis.ticks.length           : 'simpleUnit' num 2.75points
```

```
##   ..- attr(*, "unit")= int 8
##  $ axis.ticks.length.x             : NULL
##  $ axis.ticks.length.x.top         : NULL
##  $ axis.ticks.length.x.bottom      : NULL
##  $ axis.ticks.length.y             : NULL
##  $ axis.ticks.length.y.left        : NULL
##  $ axis.ticks.length.y.right       : NULL
##  $ axis.ticks.length.theta         : NULL
##  $ axis.ticks.length.r             : NULL
##  $ axis.minor.ticks.length         : 'rel' num 0.75
##  $ axis.minor.ticks.length.x       : NULL
##  $ axis.minor.ticks.length.x.top   : NULL
##  $ axis.minor.ticks.length.x.bottom: NULL
##  $ axis.minor.ticks.length.y       : NULL
##  $ axis.minor.ticks.length.y.left  : NULL
##  $ axis.minor.ticks.length.y.right : NULL
##  $ axis.minor.ticks.length.theta   : NULL
##  $ axis.minor.ticks.length.r       : NULL
##  $ axis.line                       : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x                     : NULL
##  $ axis.line.x.top                 : NULL
##  $ axis.line.x.bottom              : NULL
```

```
##  $ axis.line.y                    : NULL
##  $ axis.line.y.left               : NULL
##  $ axis.line.y.right              : NULL
##  $ axis.line.theta                : NULL
##  $ axis.line.r                    : NULL
##  $ legend.background              : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ legend.margin            : 'margin' num [1:4] 5.5points 5.5points 5.
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing               : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing.x               : NULL
##  $ legend.spacing.y               : NULL
##  $ legend.key                     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.key.size              : 'simpleUnit' num 1.2lines
##   ..- attr(*, "unit")= int 3
##  $ legend.key.height              : NULL
##  $ legend.key.width               : NULL
## $ legend.key.spacing           : 'simpleUnit' num 5.5points
##   ..- attr(*, "unit")= int 8
##  $ legend.key.spacing.x           : NULL
##  $ legend.key.spacing.y           : NULL
```

```
##  $ legend.frame                    : NULL
##  $ legend.ticks                    : NULL
##  $ legend.ticks.length             : 'rel' num 0.2
##  $ legend.axis.line                : NULL
##  $ legend.text                     :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.position            : NULL
##  $ legend.title                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
```

```
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.position           : NULL
##  $ legend.position                 : chr "right"
##  $ legend.position.inside          : NULL
##  $ legend.direction                : NULL
##  $ legend.byrow                    : NULL
##  $ legend.justification            : chr "center"
##  $ legend.justification.top        : NULL
##  $ legend.justification.bottom     : NULL
##  $ legend.justification.left       : NULL
##  $ legend.justification.right      : NULL
##  $ legend.justification.inside     : NULL
##  $ legend.location                 : NULL
##  $ legend.box                      : NULL
##  $ legend.box.just                 : NULL
## $ legend.box.margin          : 'margin' num [1:4] 0cm 0cm 0cm 0cm
```

```
##   ..- attr(*, "unit")= int 1
##  $ legend.box.background           : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing            : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##   [list output truncated]
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

```r
ggplot(df3_clean, aes(x = -log10(P))) +
  geom_histogram(bins = 50, fill = "skyblue", color = "white") +
  labs(title = "Distribution of -log10(P-values)",
       x = "-log10(P-value)", y = "Count") +xlim(2, 4)
```

```
  theme_minimal()
```

```
## List of 136
##  $ line                            :List of 6
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                            :List of 5
```

```
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                            :List of 11
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##  ..$ margin     : 'margin' num [1:4] 0points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                           : NULL
##  $ aspect.ratio                    : NULL
##  $ axis.title                      : NULL
```

```
##  $ axis.title.x                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 2.75points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
```

```
##  ..$ margin     : 'margin' num [1:4] 0points 0points 2.75points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom             : NULL
##  $ axis.title.y                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 0points 2.75points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left               : NULL
##  $ axis.title.y.right              :List of 11
##   ..$ family       : NULL
```

```
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 0points 0points 0points 2.75points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                       :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
```

```
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x                     :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 2.2points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top                 :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
```

```
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 0points 0points 2.2points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom              : NULL
##  $ axis.text.y                     :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 0points 2.2points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left                : NULL
```

```
##  $ axis.text.y.right                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 0points 0points 0points 2.2points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.theta                  : NULL
##  $ axis.text.r                      :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0.5
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
```

```
##   ..$ lineheight   : NULL
##  ..$ margin     : 'margin' num [1:4] 0points 2.2points 0points 2.2point
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                      : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.ticks.x                    : NULL
##  $ axis.ticks.x.top                : NULL
##  $ axis.ticks.x.bottom             : NULL
##  $ axis.ticks.y                    : NULL
##  $ axis.ticks.y.left               : NULL
##  $ axis.ticks.y.right              : NULL
##  $ axis.ticks.theta                : NULL
##  $ axis.ticks.r                    : NULL
##  $ axis.minor.ticks.x.top          : NULL
##  $ axis.minor.ticks.x.bottom       : NULL
##  $ axis.minor.ticks.y.left         : NULL
##  $ axis.minor.ticks.y.right        : NULL
##  $ axis.minor.ticks.theta          : NULL
##  $ axis.minor.ticks.r              : NULL
## $ axis.ticks.length           : 'simpleUnit' num 2.75points
```

```
##   ..- attr(*, "unit")= int 8
##  $ axis.ticks.length.x             : NULL
##  $ axis.ticks.length.x.top         : NULL
##  $ axis.ticks.length.x.bottom      : NULL
##  $ axis.ticks.length.y             : NULL
##  $ axis.ticks.length.y.left        : NULL
##  $ axis.ticks.length.y.right       : NULL
##  $ axis.ticks.length.theta         : NULL
##  $ axis.ticks.length.r             : NULL
##  $ axis.minor.ticks.length         : 'rel' num 0.75
##  $ axis.minor.ticks.length.x       : NULL
##  $ axis.minor.ticks.length.x.top   : NULL
##  $ axis.minor.ticks.length.x.bottom: NULL
##  $ axis.minor.ticks.length.y       : NULL
##  $ axis.minor.ticks.length.y.left  : NULL
##  $ axis.minor.ticks.length.y.right : NULL
##  $ axis.minor.ticks.length.theta   : NULL
##  $ axis.minor.ticks.length.r       : NULL
##  $ axis.line                       : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x                     : NULL
##  $ axis.line.x.top                 : NULL
##  $ axis.line.x.bottom              : NULL
```

```
##  $ axis.line.y                    : NULL
##  $ axis.line.y.left               : NULL
##  $ axis.line.y.right              : NULL
##  $ axis.line.theta                : NULL
##  $ axis.line.r                    : NULL
##  $ legend.background              : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ legend.margin            : 'margin' num [1:4] 5.5points 5.5points 5.
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing               : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing.x               : NULL
##  $ legend.spacing.y               : NULL
##  $ legend.key                     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.key.size              : 'simpleUnit' num 1.2lines
##   ..- attr(*, "unit")= int 3
##  $ legend.key.height              : NULL
##  $ legend.key.width               : NULL
## $ legend.key.spacing           : 'simpleUnit' num 5.5points
##   ..- attr(*, "unit")= int 8
##  $ legend.key.spacing.x           : NULL
##  $ legend.key.spacing.y           : NULL
```

```
##  $ legend.frame                    : NULL
##  $ legend.ticks                    : NULL
##  $ legend.ticks.length             : 'rel' num 0.2
##  $ legend.axis.line                : NULL
##  $ legend.text                     :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.position            : NULL
##  $ legend.title                    :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
```

```
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.position           : NULL
##  $ legend.position                 : chr "right"
##  $ legend.position.inside          : NULL
##  $ legend.direction                : NULL
##  $ legend.byrow                    : NULL
##  $ legend.justification            : chr "center"
##  $ legend.justification.top        : NULL
##  $ legend.justification.bottom     : NULL
##  $ legend.justification.left       : NULL
##  $ legend.justification.right      : NULL
##  $ legend.justification.inside     : NULL
##  $ legend.location                 : NULL
##  $ legend.box                      : NULL
##  $ legend.box.just                 : NULL
## $ legend.box.margin          : 'margin' num [1:4] 0cm 0cm 0cm 0cm
```

```
##   ..- attr(*, "unit")= int 1
##  $ legend.box.background           : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing            : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##   [list output truncated]
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

```r
#To create figure in slide deck
x <- seq(-4, 4, length.out = 1000)
y <- dnorm(x)

df_norm <- data.frame(x = x, y = y)

z_cutoff <- qnorm(0.95)

shade_df <- subset(df_norm, x > z_cutoff)



ggplot(df_norm, aes(x, y)) +
  geom_line(color = "black", size = 1) +  # Normal curve
```

```
  geom_area(data = shade_df, aes(x, y), fill = "skyblue", alpha = 0.6) +
  geom_vline(xintercept = z_cutoff, color = "red", linetype = "dashed",
  labs(
    title = "Standard Normal Distribution with One-Sided p = 0.05 Cutof
    x = "Z-score",
    y = "Density"
  ) +
  theme_minimal()
```

References

[1] https://www.milieuinterieur.fr/en/about-us/the-milieu-interieur/

[2]Scepanovic, P., Alanio, C., Hammer, C., et al. (2018). Human genetic variants and age are the strongest predictors of humoral immune responses to common pathogens and vaccines. Genome Medicine, 10, 59. https://doi.org/10.1186/s13073-018-0568-8

[3] https://www.rdocumentation.org/packages/qqman/versions/0.1.2/topics/manhattan

[4] https://search.r-project.org/CRAN/refmans/mutoss/html/multiple.down.html

[5] Benjamini, Y., Krieger, A. M., & Yekutieli, D. (2006). Adaptive Linear Step-up Procedures That Control the False Discovery Rate. Biometrika, 93(3), 491–507. https://doi.org/10.1093/biomet/93.3.491

[6] Storey JD, Tibshirani R. Statistical significance for genomewide studies. Proc Natl Acad Sci U S A. 2003 Aug 5;100(16):9440-5. doi: 10.1073/pnas.1530509100. Epub 2003 Jul 25. PMID: 12883005; PMCID: PMC170937.

[7] https://www.rdocumentation.org/packages/qvalue/versions/2.4.2/topics/qvalue

[8] https://www.ncbi.nlm.nih.gov/gdv/

[9] Arai S, Miyazaki T. Impaired maturation of myeloid progenitors in mice lacking novel Polycomb group protein MBT-1. EMBO J. 2005 May 18;24(10):1863-73. doi: 10.1038/sj.emboj.7600654