ATXN2Analysis

Diksha and Jinko

2024-03-10

1. Preliminaries

Load the necessary R packages.

```
library(readxl)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- 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 become errors
```

```
library(SKAT)
```

```
## Loading required package: Matrix
```

```
## Warning: package 'Matrix' was built under R version 4.3.2
```

```
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loading required package: SPAtest
## Loading required package: RSpectra
```

Rebuild the necessary part of the `persons` object from the exploratory analysis.

```
persons<-read_excel("../ATXN2persons.xlsx", sheet="Data")
persons$`Created at` <-  as.Date(persons$`Created at`)
persons$age <- year(persons$`Created at`) - year(persons$DOB)

# Adjust age for leap year cases
persons$age <- ifelse(month(persons$DOB) > month(persons$`Created at`) | (month(persons$DOB) == month(p

# Reshape the enrichment kits information
persons$`Enrichment kits` <-  factor(persons$`Enrichment kits`, levels = c("Twist Comprehensive Exome R
"Twist Comprehensive Exome plus Refseq",
 "Twist Comprehensive Exome plus Refseq, Twist Exome 2.0, Twist Mix (Comprehensive plus 2.0)", "Twist M
), labels = c("TCER vs2", "Exon v7","TCE(RefSeq)","TCE(R,Ex,Mix)", "T Mix"))
```

Rebuild the necessary part of the `variants_new` object from the exploratory data analysis.

```
variants<-read_excel("../ATXN2variants.xlsx", sheet="Sheet1")
variants_new = left_join(variants, persons, by= "SampleIndex", relationship="many-to-many") %>%
  distinct(SampleIndex, ND)
variants_new = merge(variants, variants_new, by = "SampleIndex")

variants_new <- variants_new %>%
  select(-c(SampleIndex, SampleNumber)) %>%
  select(cDNA, everything())
variants_new <-variants_new %>%
  distinct(cDNA, .keep_all = TRUE)
variants_new$ND <- NULL; variants_new$Gene <- NULL; variants_new$OMIM <-NULL
```

2. Overview

We conduct analyses on two different sets of data. The first dataset retains those subjects who may have neurodegenerative disease, the `maybe` ND subjects, and groups them with the subjects who have neurodegenerative disease (ND), the `yes` ND subjects. The `maybe` and `yes` ND subjects are viewed as cases while the `no` ND subjects are viewed as controls. We refer to this first set of data as the "Dataset retaining `maybe` ND subjects". The second set of data discards the `maybe` ND subjects and keeps only the `yes` ND subjects as cases and the `no` ND subjects as controls. We refer to this smaller set of data as the "Dataset removing `maybe` ND subjects".

Throughout this document we apply methods from the `SKAT` package in R (Lee S, Zhao Z, Miropolsky wcfL, Wu M (2023). *SKAT: SNP-Set (Sequence) Kernel Association Test*. R package version 2.2.5, https://CRAN.R-project.org/package=SKAT). The SKAT methodology implements a score test of the aggregated effects of rare genetic variants in some genomic region of interest, such as exon 1 of the ATXN2 gene. Briefly, a generalized linear model involving only non-genetic covariates X is fitted to the response variable Y; this is the so-called null model. Possible null models include logistic regression for independent (unrelated) binary responses and Gaussian regression for continuous responses related through a kinship matrix K. Once a null model is fit, a score test is performed including the effects encoded in the genetic variants matrix Z and investigator-specified weights for these variants. Typically, weights are based on the population allele frequencies of the genetic variants, with rare variants getting more weight in the analysis.

3. Dataset retaining `maybe` ND subjects

The first dataset that we look at retains the `maybe` ND subjects, and groups them with the `yes` ND subjects. The `maybe` and `yes` ND subjects are viewed as cases while the `no` ND subjects are viewed as controls.

3.1 Covariates matrix (X)

We start by setting up the non-genetic covariates, X, for our null model. Non-genetic covariates in our study include the potential confounding variables age, sex and enrichment kit. Based on investigator advice, we have grouped the enrichment kits coded as `TCE(R,Ex,Mix)` (3 subjects) and `T Mix` (32 subjects) and kept the enrichment kit coded as `Exon v7` (4 subjects) separate from the others as it is sourced from a different vendor. The enrichment kit used for the largest number of subjects is coded as `TCER vs2` (240 subjects) and will be used as a baseline category in all our regression analyses.

```
table(persons$`Enrichment kits`)
```

```
##
##      TCER vs2       Exon v7   TCE(RefSeq) TCE(R,Ex,Mix)         T Mix
##           240             4            79             3            32
```

```
covariate.ekits1 =  ifelse(persons$`Enrichment kits` == "TCE(RefSeq)", 1, 0)
covariate.ekits2 = ifelse(persons$`Enrichment kits` == "TCE(R,Ex,Mix)" | persons$`Enrichment kits` == "
covariate.ekits3 = ifelse(persons$`Enrichment kits` == "Exon v7", 1, 0)

X = tibble(persons$sampleID, covariate.ekits1, covariate.ekits2, covariate.ekits3, persons$age, persons
names(X) <- c("sampleID", "ekit.TCE", "ekit.TwistMix", "ekit.Exon", "age", "sex")
```

3.2 Phenotype vector (Y)

Next we set up the response or phenotype vector, Y. The response variable Y is coded as 1 for ND `yes` and `maybe`, and as 0 for ND `no`.

```
Y = persons$ND>0
table(Y)
```

```
## Y
## FALSE  TRUE
##   161   197
```

3.3 Kinship matrix (K)

The original data contain four parent-child trios, a father-son duo and a sister-pair. All these relative clusters involve individuals without neurodegenerative disease. Let's set up the kinship matrix.

```
# Initialize kinship matrix with all zeros
K <- diag(rep(0.5, nrow(persons)))
rownames(K) <- persons$sampleID
colnames(K) <- persons$sampleID
```

```
# Modify kinship matrix for relatives
K["P-MY120", "P-MY121"] <- 0.25
K["P-MY121", "P-MY120"] <- 0.25
K["EX457", "P-SY163"] <- 0.25
K["P-SY163",  "EX457"] <- 0.25
K["EX458", "P-SY163"] <-0.25
K["P-SY163","EX458"] <-0.25
K["EX461", "P-SY165"] <- 0.25
K["P-SY165","EX461"] <- 0.25
K["EX462", "P-SY165"] <- 0.25
K["P-SY165","EX462"] <- 0.25
K["EX463", "P-SY166"] <- 0.25
K[ "P-SY166", "EX463"] <- 0.25
K["EX464", "P-SY166"] <- 0.25
K["P-SY166","EX464"] <- 0.25
K["P-SY169","EX470"] <- 0.25
K["EX470", "P-SY169"] <- 0.25
K["EX471", "P-SY169"] <- 0.25
K[ "P-SY169","EX471"] <- 0.25
K["P-MY118", "P-MY119"] <- 0.25
K["P-MY119","P-MY118"] <- 0.25
```

3.4 Genotype matrix (Z)

Now set up the genotype matrix, Z. Code Z so that each row is a subject in the study and each column is a rare variant observed in the study.

```
Z = matrix(0, nrow = nrow(persons), ncol = nrow(variants_new))
rownames(Z) = persons$sampleID
colnames(Z) = variants_new$cDNA

for (cDNA_value in colnames(Z)) {
    rows_to_update <- persons$`ATXN2ex1 variant1` == cDNA_value

  Z[rows_to_update, cDNA_value] <- 1
}

for (cDNA_value in colnames(Z)) {
    rows_to_update <- persons$`ATXN2ex1 variant2` == cDNA_value

  Z[rows_to_update, cDNA_value] <- 1
}
```

3.5 Null models

All the null models consider only the potential confounding variables, X, as covariates; they do not consider the genetic variant information, Z. Based on investigator input and the results of our earlier exploratory analysis of the data, the potential confounding variables X are the variables age, sex and enrichment kit. Two types of SKAT null models can be applied. The first model accounts for a binary response variable, Y, by fitting a logistic regression but assumes unrelated subjects. The second null model accounts for related subjects through a kinship matrix, K, but assumes a Gaussian response variable, Y.

3.5.1 Logistic regression

We start with a SKAT null model that accounts for the binary response through a logistic regression, but takes only unrelated (i.e. independent) subjects. For this, we need to remove the non-ND children P-SY163, P-SY165, P-SY166 and P-SY169 in the four case-parent trios. We will also remove the son from the father-son duo and the younger of the two siblings in the sibling pair. All these relative clusters involve individuals who do not have a neurodegenerative disease. We keep the older individuals in these clusters so that our non-ND "controls" have ages which better match the older ages of the ND "cases" in our study.

```
removekids<-c("P-SY163", "P-SY165", "P-SY166", "P-SY169", "P-MY119", "P-MY120")
rem<-X$sampleID %in% removekids
Xsub<-X[!rem,]
Ysub<-Y[!rem]
Zsub<-Z[!rem,]

null_model_binary<-SKAT_Null_Model(Ysub ~ age + sex + ekit.TCE + ekit.TwistMix + ekit.Exon, data=Xsub, 
```

```
## Sample size (non-missing y and X) = 352, which is < 2000. The small sample adjustment is applied!
```

The functions for fitting null models in SKAT do not allow us to see what the effects of the covariates are in X. Since the logistic regression assumes independent (i.e. unrelated) subjects, we can equivalently call the `glm()` function in R to see what the effects of age, gender, and enrichment kit are.

```
obj<-glm(Ysub ~ age + sex + ekit.TCE + ekit.TwistMix + ekit.Exon, family=binomial, data=Xsub)
summary(obj)
```

```
##
## Call:
## glm(formula = Ysub ~ age + sex + ekit.TCE + ekit.TwistMix + ekit.Exon,
##     family = binomial, data = Xsub)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.101130   0.425497  -7.288 3.14e-13 ***
## age            0.076831   0.008029   9.569  < 2e-16 ***
## sexMale        0.079097   0.284635   0.278   0.7811
## ekit.TCE      -0.846028   0.343175  -2.465   0.0137 *
## ekit.TwistMix  0.547541   0.521103   1.051   0.2934
## ekit.Exon      0.524046   1.469109   0.357   0.7213
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 482.95  on 351  degrees of freedom
## Residual deviance: 309.92  on 346  degrees of freedom
## AIC: 321.92
##
## Number of Fisher Scoring iterations: 5
```

The term `ekit.TCE` corresponds to a combination of two enrichment kits in the original `persons.xls` excel file received from the investigator: `Twist Comprehensive Exome plus Refseq, Twist Exome 2.0, Twist`

`Mix (Comprehensive plus 2.0)` (3 subjects) and `Twist Mix (Comprehensive plus 2.0)` (32 subjects). Age and `ekit.TCE` have significant terms in the logistic regression. Age is positively associated with neurodegenerative disease whereas `ekit.TCE` is negatively associated relative to the other enrichment kits. The significant negative association for `ekit.TCE` suggests that patients who don't have neurodegenerative disease tend to be assigned to one of these two enrichment kits. This negative association is hard to explain given that the samples are supposed to be randomized to enrichment kits in the lab.

3.5.2 Gaussian regression

Now we fit another SKAT null model that assumes a continuous Gaussian (rather than binary) response but accounts for the relationships amongst *all* subjects through the kinship matrix, K.

```
null_model_cts <- SKAT_NULL_emmaX(Y ~ age + sex + ekit.TCE + ekit.TwistMix + ekit.Exon, K = K, data=X)
```

Unfortunately, the `glm()` function assumes the subjects are unrelated and so cannot be called for an equivalent Gaussian regression.

3.6 Alternative models

At this stage, we consider the alternative model that includes the genetic variants coded in Z.

3.6.1 Questionable variants

The investigator has asked us to set to missing three questionable variants, `c.42del` and `c.80_85del`, `c.39_40del` that are highlighted in yellow in the original `variants` excel spreadsheet. To eliminate these variants from the SKAT analysis, we will specify a sample frequency threshold for ignoring variants. What are the sample frequencies of these questionable variants in our data?

```
sum(Z[,"c.42del"])/length(Z[,"c.42del"])
```

```
## [1] 0.02513966
```

```
sum(Z[,"c.80_85del"])/length(Z[,"c.80_85del"]) #0.0056
```

```
## [1] 0.005586592
```

```
sum(Z[,"c.39_40del"])/length(Z[,"c.39_40del"])
```

```
## [1] 0.01955307
```

```
Z[Z[,"c.42del"]>0,"c.42del"]<-NA
Z[Z[,"c.80_85del"]>0,"c.80_85del"]<-NA
Z[Z[,"c.39_40del"]>0,"c.39_40del"]<-NA

#Do the same for Zsub
sum(Zsub[,"c.42del"])/length(Zsub[,"c.42del"])
```

```
## [1] 0.02556818
```

```
sum(Zsub[,"c.80_85del"])/length(Zsub[,"c.80_85del"])
```

```
## [1] 0.005681818
```

```
sum(Zsub[,"c.39_40del"])/length(Zsub[,"c.39_40del"])
```

```
## [1] 0.01988636
```

```
Zsub[Zsub[,"c.39_40del"]>0,"c.39_40del"]<-NA
Zsub[Zsub[,"c.42del"]>0,"c.42del"]<-NA
Zsub[Zsub[,"c.80_85del"]>0,"c.80_85del"]<-NA
```

The variant `c.42del` has an observed frequency of about 0.025 in the data, the variant `c.80_85del` has a frequency of about 0.0056, and the variant `c.39_40del` has a frequency of about 0.020. We should therefore set the `SKAT()` function argument `missing_cutoff` to slightly lower than the lowest of these, say to 0.005. Setting the cutoff in this way ensures that the `SKAT()` function will ignore these three questionable variants in the analysis.

3.6.2 Variant weights

In the SKAT analysis, each of the variants will be weighted according to its allele frequency in the population. The lower the population allele frequency of a variant, the more weight it is given. Lower frequency variants are given more weight than higher frequency variants because population-genetics theory predicts they are more likely to be under negative selection and therefore deleterious. For the variant weights, we will use the population allele frequencies in the publicly-available gnomAD database.

First, let's look at how many variants we're dealing with:

```
variants_new[,c("cDNA", "GnomVariantFrequency", "GnomAlleleNumber")]
```

```
##                                    cDNA  GnomVariantFrequency GnomAlleleNumber
## 1                          c.176_190dup 2.5139999999999999E-4           111366
## 2                            c.39_40del              1.506E-3               NA
## 3                               c.60del 3.6982999999999999E-3           141686
## 4                            c.57_80del 5.2320000000000003E-4           135706
## 5                            c.80_85del 4.0030000000000003E-4               NA
## 6                            c.42_58del 1.3408999999999999E-3           140202
## 7                               c.42del 2.7476000000000002E-3               NA
## 8           c.74_75insACAGCAGCAGCAGCA              4.751E-4           141018
## 9        c.71_72insACAGCAGCAGCAGCAGCA              1.062E-3           141240
## 10 c.83_84insACAGCAGCAGCAGCAGCAGCAGCA 1.8980000000000001E-4           126468
## 11                    c.65_66insACAGCA 5.9559999999999995E-4           142710
## 12             c.71_72insACAGCAGCAGCA 7.4625000000000004E-3           141240
## 13                          c.57_58insG 5.4049999999999996E-4           140606
## 14                           c.68_85del         not in GnomAD            72282
## 15                         c.56_57insGC 4.0840000000000001E-4            26932
## 16                           c.54_58del 4.9930000000000005E-4           140202
## 17                           c.57_59del 2.2796000000000001E-3           133358
## 18                           c.48_58del 8.7020000000000001E-4           140202
## 19                           c.51_56dup         not in GnomAD            26932
```

We're dealing with 19 variants. Two of these variants are `not in GnomAD`; i.e., they have missing allele frequencies in the gnomAD database because they were not observed in a small sample of sizes 26932 and 72282 allelic copies, respectively (see the `GnomAlleleNumber` column above). We'll impute their allele frequencies to be half the minimum gnomAD allele frequency for the variants observed in the study.

```
allele_freq <- variants_new$GnomVariantFrequency

# Converting non-numeric values to NA
allele_freq[allele_freq == "not in GnomAD"] <- NA
allele_freq= as.numeric(allele_freq)
min_freq <- min(as.numeric(allele_freq), na.rm = TRUE)
# Replacing NA values with half of the minimum allele frequency
allele_freq[is.na(allele_freq)] <- (min_freq / 2)
```

Next, calculate the weights for the SKAT function using the allele frequencies.

```
weights <- 1 / sqrt(allele_freq *(1 - allele_freq))
```

3.6.3 Logistic regression

Now we're ready to test with SKAT using the null model for a binary outcome and the subsetted Xsub, Ysub, Zsub matrices for unrelated subjects.

```
skat_result_binary <- SKAT(Z=as.matrix(Zsub), obj=null_model_binary,  method = "optimal.adj", weights =
```

```
## Warning: 3 SNPs with either high missing rates or no-variation are excluded!
```

```
# Show SKAT test results
skat_result_binary$p.value
```

```
## [1] 0.02737104
```

Rare variants in exon 1 of the ATXN2 gene are associated with ND status (yes and maybe versus no; $p = .03$) in a logistic regression analysis of ND status as a binary response. The logistic regression analysis is based on $n = 352$ unrelated subjects and adjusts for age, sex and enrichment kit as potential confounding variables.

3.6.4 Gaussian regression

Next test the association with SKAT using the null model for a continuous Gaussian response and accounting for the related subjects in the X, Y, Z matrices.

```
skat_result_cts <- SKAT(Z=Z, obj=null_model_cts,  method = "optimal.adj", weights = weights, is_dosage="
```

```
## Warning: 3 SNPs with either high missing rates or no-variation are excluded!
```

```
# Show SKAT test results
skat_result_cts$p.value
```

```
## [1] 0.02117829
```

Rare variants in exon 1 of the ATXN2 gene continue to be significantly associated with ND status (yes and maybe versus no; $p = .02$), in a regression analysis of ND status as a Gaussian response. The regression analysis is based on $n = 358$ related subjects and adjusts for age, sex and enrichment kit as potential confounding variables and the relatedness of four parent-child trios, a father-son duo, and a sibling pair.

4. Dataset removing maybe ND subjects

Set up a new phenotype vector, Y.

```
persons_without_maybes = persons[persons$ND != 3, ]
Y = persons_without_maybes$ND==1
table(Y)
```

```
## Y
## FALSE  TRUE
##   161   134
```

```
length(Y) #295 persons without maybe for ND status
```

```
## [1] 295
```

Set up a new covariates matrix, X.

```
covariate.ekits1 =  ifelse(persons_without_maybes$`Enrichment kits` == "TCE(RefSeq)", 1, 0)
covariate.ekits2 = ifelse(persons_without_maybes$`Enrichment kits` == "TCE(R,Ex,Mix)" | persons_without_
covariate.ekits3 = ifelse(persons_without_maybes$`Enrichment kits` == "Exon v7", 1, 0)

X = tibble(persons_without_maybes$sampleID, covariate.ekits1, covariate.ekits2, covariate.ekits3, perso
names(X) <- c("sampleID", "ekit.TCE", "ekit.TwistMix", "ekit.Exon", "age", "sex")
```

Set up a new genotypes matrix, Z.

```
Z = matrix(0, nrow = nrow(persons_without_maybes), ncol = nrow(variants_new))
rownames(Z) = persons_without_maybes$sampleID
colnames(Z) = variants_new$cDNA

for (cDNA_value in colnames(Z)) {
    rows_to_update <- persons_without_maybes$`ATXN2ex1 variant1` == cDNA_value

  Z[rows_to_update, cDNA_value] <- 1
}

for (cDNA_value in colnames(Z)) {
    rows_to_update <- persons_without_maybes$`ATXN2ex1 variant2` == cDNA_value

  Z[rows_to_update, cDNA_value] <- 1
}
```

Remove the individuals who are the younger relatives of others in the study for the logistic regression analysis. (Note: Everyone who is in a relatedness cluster in the study has non-ND status.)

```
removekids<-c("P-SY163", "P-SY165", "P-SY166", "P-SY169","P-MY120", "P-MY119" )
rem<-X$sampleID %in% removekids
Xsub<-X[!rem,]
Ysub<-Y[!rem]
Zsub<-Z[!rem,]
```

Fit the null model for a logistic regression in SKAT.

```
null_model_binary<-SKAT_Null_Model(Ysub ~ age + sex + ekit.TCE + ekit.TwistMix + ekit.Exon, data=Xsub, 
```

```
## Sample size (non-missing y and X) = 289, which is < 2000. The small sample adjustment is applied!
```

Fit the null model with a logistic regression in glm to be able to see the significance of the various covariate effects.

```
obj<-glm(Ysub ~ age + sex + ekit.TCE + ekit.TwistMix + ekit.Exon, family=binomial, data=Xsub)
summary(obj)
```

```
##
## Call:
## glm(formula = Ysub ~ age + sex + ekit.TCE + ekit.TwistMix + ekit.Exon,
##     family = binomial, data = Xsub)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.83478    0.53798  -7.128 1.02e-12 ***
## age            0.08876    0.01051   8.442  < 2e-16 ***
## sexMale       -0.38272    0.33158  -1.154  0.24840
## ekit.TCE      -1.21687    0.40482  -3.006  0.00265 **
## ekit.TwistMix  0.31758    0.60489   0.525  0.59956
## ekit.Exon      1.08596    1.62439   0.669  0.50379
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 399.11  on 288  degrees of freedom
## Residual deviance: 237.35  on 283  degrees of freedom
## AIC: 249.35
##
## Number of Fisher Scoring iterations: 5
```

As in the analysis including the ND `maybe` subjects, the two enrichment kits coded as`ekit.TCE` are negatively associated with ND status, relative to the other enrichment kits.

Next set up the kinship matrix K for the regression analysis that (incorrectly) treats the binary ND status as a continuous Gaussian variable but can account for relatedness.

```
# Creating a kinship matrix with all zeros
K <- diag(rep(0.5, nrow(persons_without_maybes)))
rownames(K) <- persons_without_maybes$sampleID
colnames(K) <- persons_without_maybes$sampleID

#kinship matrix
K["P-MY120", "P-MY121"] <- 0.25
K["P-MY121", "P-MY120"] <- 0.25
K["EX457", "P-SY163"] <- 0.25
K["EX458", "P-SY163"] <-0.25
K["EX461", "P-SY165"] <- 0.25
```

```
K["EX462", "P-SY165"] <- 0.25
K["EX463", "P-SY166"] <- 0.25
K["EX464", "P-SY166"] <- 0.25
K["EX470", "P-SY169"] <- 0.25
K["EX471", "P-SY169"] <- 0.25
K["P-MY118", "P-MY119"] <- 0.25
K["P-SY163",  "EX457"] <- 0.25 #JG note: should be EX457 and P-SY163
#Set the non-zero lower-diagonal entries by loop
for(i in 1:(nrow(persons_without_maybes)-1))
  for(j in (i+1):nrow(persons_without_maybes))
    K[i,j]<- K[j,i]
```

The investigator has asked us to set to missing three questionable variants, `c.42del` and `c.80_85del`, `c.39_40del` that are highlighted in yellow in variants spreadsheet. What are the observed frequencies of these questionable variants in the dataset that removes the ND `maybe` subjects?

```
sum(Z[,"c.42del"])/length(Z[,"c.42del"])
```

```
## [1] 0.02372881
```

```
sum(Z[,"c.80_85del"])/length(Z[,"c.80_85del"]) #0.00678
```

```
## [1] 0.006779661
```

```
sum(Z[,"c.39_40del"])/length(Z[,"c.39_40del"])
```

```
## [1] 0.02033898
```

```
Z[Z[,"c.42del"]>0,"c.42del"]<-NA
Z[Z[,"c.80_85del"]>0,"c.80_85del"]<-NA
Z[Z[,"c.39_40del"]>0,"c.39_40del"]<-NA
#Do the same for Zsub
sum(Zsub[,"c.42del"])/length(Zsub[,"c.42del"])
```

```
## [1] 0.02422145
```

```
sum(Zsub[,"c.80_85del"])/length(Zsub[,"c.80_85del"]) #0.00692
```

```
## [1] 0.006920415
```

```
sum(Zsub[,"c.39_40del"])/length(Zsub[,"c.39_40del"])
```

```
## [1] 0.02076125
```

```
Zsub[Zsub[,"c.39_40del"]>0,"c.39_40del"]<-NA
Zsub[Zsub[,"c.42del"]>0,"c.42del"]<-NA
Zsub[Zsub[,"c.80_85del"]>0,"c.80_85del"]<-NA
```

The variant `c.42del` has an observed frequency of about 0.024 in the data, the variant `c.80_85del` has a frequency of about 0.0067 and the variant `c.39_40del` has a frequency of about 0.02. We should therefore set the `SKAT()` function argument `missing_cutoff` to slightly lower than the lowest of these, say to 0.005. Setting the cutoff in this way ensures that the `SKAT()` function will discard these three questionable variants in the analysis.

```
skat_result_binary <- SKAT(Z=as.matrix(Zsub), obj=null_model_binary,  method = "optimal.adj", weights =
```

```
## Warning: 5 SNPs with either high missing rates or no-variation are excluded!
```

```
nrow(Zsub) #number of subjects in the analysis
```

```
## [1] 289
```

```
apply(Zsub,2,sum)
```

```
##                        c.176_190dup                            c.39_40del
##                                   1                                    NA
##                            c.60del                            c.57_80del
##                                   7                                     1
##                          c.80_85del                            c.42_58del
##                                  NA                                     4
##                           c.42del          c.74_75insACAGCAGCAGCAGCA
##                                  NA                                     1
##       c.71_72insACAGCAGCAGCAGCAGCA c.83_84insACAGCAGCAGCAGCAGCAGCAGCA
##                                   1                                     1
##                   c.65_66insACAGCA             c.71_72insACAGCAGCAGCA
##                                   1                                     6
##                         c.57_58insG                            c.68_85del
##                                   1                                     1
##                        c.56_57insGC                            c.54_58del
##                                   1                                     0
##                          c.57_59del                            c.48_58del
##                                   0                                     3
##                          c.51_56dup
##                                   2
```

The reason for the warning message about five (rather than the expected three variants that we set to be missing) is that two additional variants have been dropped from the analysis because they are present only in subjects with ND status `maybe`. As can be seen from the above output, they are both deletions: `c.54_58del` and `c.57_59del`.

Finally we show the results of our association test. # Show SKAT test results

```
skat_result_binary$p.value
```

```
## [1] 0.00738595
```

When subjects who are `maybe` ND are excluded, rare variants in exon 1 of the ATXN2 gene are associated with ND status (i.e. `yes` versus `no`; $p = 0.005 - 0.008$) in a logistic regression analysis. The logistic regression analysis is based on $n = 289$ unrelated subjects and adjusts for age, sex and enrichment kit as potential confounding variables.