1-ProjectDoc

2022-11-04

Goup work 1

File SNPdata * tab delimited * chromosome_position as row names * rows contains 0 for AA, 1 for Aa, 2 for aa (0 or 1 or 2 mutated alleles) * patient_id as columns * in column -> 1:1200 are patients, 1201:2000 are control subjects

qcalculation

1. Take in input numerical matrix in SNPdata like format
2. Calculate MAF `q` for each SNP (chromosome position)
3. return maf vector

```
qcalculation <- function(SNPdata) {
  calcq <- function(d) {
    N = length(d)
    AA = length(d[d == 0])
    Aa = length(d[d == 1])
    aa = length(d[d == 2])
    q = (aa * 2 + Aa) / (2 * N)
    return(q)
  }

  out = as.data.frame(apply(SNPdata, 1, calcq)) # apply calcq to rows
  colnames(out)="MAF"
  return(out)
}
```

HWEtest

1. Take in input numerical matrix in SNPdata like format
2. Compute a HWE test for each SNP given as input by calculating the χ^2_{obs} from the data
   - $\chi^2_{obs} = \sum_{i\in\{AA,Aa,aa\}} \frac{O_i - Np_i}{N}$ where $p_{AA} = p^2, p_{Aa} = 2pq, p_{aa} = q^2$ and O_i are the number of observed AA,Aa,aa from the provided data.
   - By computing the p value using the function `pchisq` (DO NOT use directly the `chisq.test()` function)
     - `pchisq(chi_squared, 1, lower.tail = FALSE)`
3. Return the HWE test p-values for each SNP as a vector of numeric values with names corresponding to the SNP IDs with the same order they had in the input matrix

```r
HWEtest <- function(SNPdata) {
  calcp<- function(d) {
    N = length(d)

    ## Calculate observed values -------------------------------------------
    AA = length(d[d == 0])
    Aa = length(d[d == 1])
    aa = length(d[d == 2])
    O = c(AA, Aa, aa)

    ## Expected frequencies ------------------------------------------------
    q = (aa * 2 + Aa) / (2 * N)
    p = 1 - q
    prob_expec=c(p^2,2*p*q,q^2)

    ## ChiSquared ----------------------------------------------------------
    chi_squared=sum((O-N*prob_expec)^2/(N*prob_expec))

    ## pvalue --------------------------------------------------------------
    pvalue <- pchisq(chi_squared, 1, lower.tail = FALSE)

    return(pvalue)
  }

  out = as.data.frame(apply(SNPdata, 1, calcp))
  colnames(out)=c("pvalue")
  return(out)
}
```

VARIANTanalysis

1. take as input
   - `filepath` of file with SNPdata like format
   - `indCTRL` vector indicating which column has data from control subjects
   - `MAFth` treshold to filter SNPs lower MAF, default 0.01
   - `HWEalpha` significance level alpha to be used to filter SNP with lower p-values because possibly not in HWE
2. read the file and analyse variants
   - filter out the one with MAF $<$ MAFth $\vee$ HWE-p-value $<$ HWEalpha
     - `SNPdata = SNPdata[(qcalculation(SNPdata)>MAFth & HWEtest(SNPdata)[2]>HWEalpha) , ]`
   - calculate χ^2_{obs}
     - use HWEtest, applied to control only (as in suggestion)
   - calculate p-values
     - use HWEtest, applied to control only (as in suggestion)
3. Compute q-value for each SNP using Benjamini-Hockberg procedure
   - $qvalue_i = pvalue_i \times \frac{rank_i}{N}, \quad i \in 1, N$, $rank_i$ equal to position of variant in list ordered by pvalue
4. Return matrix with data `AA_ctrl, Aa_ctrl, aa_ctrl, AA_case, Aa_case, aa_case, pval, qval`.

```
VARIANTanalysis <-
  function(filepath,
           indCTRL,
           MAFth = 0.01,
           HWEalpha = 0.01) {
  ## Input and parameters setup ----------------------------------------------
  SNPdata <- read.table("SNPdata.txt", header = TRUE, sep = "\t")
  SNPdata = SNPdata[qcalculation(SNPdata[,indCTRL]) > MAFth &
                    HWEtest(SNPdata[,indCTRL]) > HWEalpha, ]
  `%notin%` <- Negate(`%in%`)
  N = dim(SNPdata)[2]
  indPATI = 1:N
  indPATI = indPATI[indPATI %notin% indCTRL]

  ## Function definitions
  calcallele <- function(d, indCTRL, indPATI) {
    AA = length(d[d == 0])
    Aa = length(d[d == 1])
    aa = length(d[d == 2])
    O = c(AA, Aa, aa)
    return(O)
  }

  calcchip<- function(d) {
    N = length(d)
    O <- calcallele(d)
    q = (O[3]* 2 + O[2]) / (2 * N)
    p = 1 - q
    prob_expec=c(p^2,2*p*q,q^2)
    chi_squared=sum((O-N*prob_expec)^2/(N*prob_expec))
    pvalue <- pchisq(chi_squared, 2, lower.tail = FALSE)
    return(c(chi_squared, pvalue))
  }


  ## Calculations ------------------------------------------------------------
  O = as.data.frame(cbind(                            #observed data
        t(apply(SNPdata[, indCTRL], 1, calcallele)),
        t(apply(SNPdata[, indPATI], 1, calcallele))
      ))

  t=as.data.frame(t((apply(SNPdata[,indCTRL],1, calcchip))))
  chisquared=t[,1]                                    # chisquared
  pvalues = t[,2]                                     # pvalues

  ### calculate q                                     # qvalues
  r = order(unlist(pvalues), decreasing = FALSE)
  #qvalues = pvalues * N / r
  qvalues = pvalues * r / N # statistical book alpha*rank/total

  ## Final Output
  out = as.data.frame(cbind(O, pvalues, qvalues))
  colnames(out) <-c("AA_ctrl","Aa_ctrl","aa_ctrl","AA_case","Aa_case","aa_case","pval","qval")
  return(as.data.frame(out))
}
```

Tests

```
SNPdata <- read.table("SNPdata.txt", header = TRUE, sep = "\t")
vec_maf <- qcalculation(SNPdata = SNPdata)
vec_HWE_chi_pvalue <- HWEtest(SNPdata)
vec_VarAnal = VARIANTanalysis("SNPdata.txt", 1201:2000, MAFth = 0.05)
```