main

August 4, 2021

1 Differential Expression with limma-voom pipeline

1.1 Load Libraries

[1]:
```
suppressMessages({library(SummarizedExperiment)
                  library(variancePartition)
                  library(BiocParallel)
                  library(data.table)
                  library(disk.frame)
                  library(tidyverse)
                  library(memoise)
                  library(limma)
                  library(edgeR)
                  library(glue)
                  library(sva)})
```

1.2 Define functions

1.2.1 Simple functions

[2]:
```
set_parallel <- function(feature){
    parallel_param = list("genes"=32, "transcripts"=16,
                          "exons"=10, "junctions"=12)
    param = SnowParam(parallel_param[[feature]],
                      "SOCK", progressbar=FALSE)
    register(param)
}


load_counts <- function(region, feature){
    if(region == 'caudate'){
        counts_lt = list("genes"=paste0("/ceph/projects/v4_phase3_paper/inputs/
 ↪counts/_m/",
                                        ␣
 ↪"caudate_brainseq_phase3_hg38_rseGene_merged_n464.rda"),
                         "transcripts"=paste0("/ceph/projects/v4_phase3_paper/
 ↪inputs/counts/_m/",
```

```
                                                  ␣
↪"caudate_brainseq_phase3_hg38_rseTx_merged_n464.rda"),
                        "exons"=paste0("/ceph/projects/v4_phase3_paper/inputs/
↪counts/_m/",
                                                  ␣
↪"caudate_brainseq_phase3_hg38_rseExon_merged_n464.rda"),
                        "junctions"=paste0("/ceph/projects/v4_phase3_paper/
↪inputs/counts/_m/",
                                                  ␣
↪"caudate_brainseq_phase3_hg38_rseJxn_merged_n464.rda"))
       tx_file = "/ceph/projects/v4_phase3_paper/inputs/counts/_m/
↪transcripts_counts/caudate_counts.txt"
   } else if(region == 'dlpfc'){
       counts_lt = list("genes"=paste0("/ceph/projects/v4_phase3_paper/inputs/
↪counts/_m/",
                                                  ␣
↪"dlpfc_ribozero_brainseq_phase2_hg38_rseGene_merged_n453.rda"),
                        "transcripts"=paste0("/ceph/projects/v4_phase3_paper/
↪inputs/counts/_m/",
                                                  ␣
↪"dlpfc_ribozero_brainseq_phase2_hg38_rseTx_merged_n453.rda"),
                        "exons"=paste0("/ceph/projects/v4_phase3_paper/inputs/
↪counts/_m/",
                                                  ␣
↪"dlpfc_ribozero_brainseq_phase2_hg38_rseExon_merged_n453.rda"),
                        "junctions"=paste0("/ceph/projects/v4_phase3_paper/
↪inputs/counts/_m/",
                                                  ␣
↪"dlpfc_ribozero_brainseq_phase2_hg38_rseJxn_merged_n453.rda"))
       tx_file = "/ceph/projects/v4_phase3_paper/inputs/counts/_m/
↪transcripts_counts/dlpfc_counts.txt"
   } else {
       counts_lt = list("genes"=paste0("/ceph/projects/v4_phase3_paper/inputs/
↪counts/_m/",
                                                  ␣
↪"hippo_brainseq_phase2_hg38_rseGene_merged_n447.rda"),
                        "transcripts"=paste0("/ceph/projects/v4_phase3_paper/
↪inputs/counts/_m/",
                                                  ␣
↪"hippo_brainseq_phase2_hg38_rseTx_merged_n447.rda"),
                        "exons"=paste0("/ceph/projects/v4_phase3_paper/inputs/
↪counts/_m/",
                                                  ␣
↪"hippo_brainseq_phase2_hg38_rseExon_merged_n447.rda"),
```

```
                        "junctions"=paste0("/ceph/projects/v4_phase3_paper/
↪inputs/counts/_m/",
                                                                  ␣
↪"hippo_brainseq_phase2_hg38_rseJxn_merged_n447.rda"))
       tx_file = "/ceph/projects/v4_phase3_paper/inputs/counts/_m/
↪transcripts_counts/hippo_counts.txt"
    }
    return(list("count_file"=counts_lt[[feature]], "tx_file"=tx_file))
}


get_model <- function(){
    covs = c('Dx', 'Age', 'rRNA_rate', 'totalAssignedGene', 'RIN',
             'ERCCsumLogErr', 'overallMapRate', 'mitoRate', 'snpPC1',
             'snpPC2', 'snpPC3', 'Sex', 'Region', 'BrNum')
    adjust_covs = covs[1:11]
    random_effect = covs[14]
    formula <- glue("~ Sex * Region + ",
                    glue_collapse(adjust_covs, sep = " + ")) %>%
        glue(" + (1|{random_effect})")
    return(formula)
}


get_null <- function(){
    covs = c('Dx', 'Age', 'rRNA_rate', 'totalAssignedGene', 'RIN',
             'ERCCsumLogErr', 'overallMapRate', 'mitoRate', 'snpPC1',
             'snpPC2', 'snpPC3', 'Sex', 'Region', 'BrNum')
    adjust_covs = covs[1:11]
    random_effect = covs[14]
    null_form <- glue("~ ", glue_collapse(adjust_covs[2:11], sep = " + ")) %>%
      glue(" + (1|{random_effect}) + (1|Dx)")
    return(null_form)
}


save_volcanoPlot <- function(top, label, feature){
    pdf(file=paste0(feature, "/volcanoPlot_", label, ".pdf"), 8, 6)
    with(top, plot(logFC, -log10(P.Value), pch=20, cex=0.6))
    with(subset(top, adj.P.Val<=0.05), points(logFC, -log10(P.Value),
                                              pch=20, col='red', cex=0.6))
    with(subset(top, abs(logFC)>0.50), points(logFC, -log10(P.Value),
                                              pch=20, col='orange', cex=0.6))
    with(subset(top, adj.P.Val<=0.05 & abs(logFC)>0.50),
         points(logFC, -log10(P.Value), pch=20, col='green', cex=0.6))
    dev.off()
}
```

```
save_MAplot <- function(top, label, feature){
    pdf(file=paste0(feature, "/MAplot_", label, ".pdf"), 8, 6)
    with(top, plot(AveExpr, logFC, pch=20, cex=0.5))
    with(subset(top, adj.P.Val<0.05),
         points(AveExpr, logFC, col="red", pch=20, cex=0.5))
    dev.off()
}


extract_de <- function(contrast, label, fit, feature){
    top <- topTable(fit, coef=contrast, number=Inf, sort.by="P")
    top <- top[order(top$P.Value), ]
    top.fdr <- top %>% filter(adj.P.Val<=0.05)
    print(paste("Comparison for:", label))
    print(paste('There are:', dim(top.fdr)[1], 'DE features!'))
    fwrite(top,
           file=paste0(feature, "/diffExpr_", label, "_full.txt"),
           sep='\t', row.names=TRUE)
    fwrite(top.fdr,
           file=paste0(feature, "/diffExpr_", label, "_FDR05.txt"),
           sep='\t', row.names=TRUE)
    save_volcanoPlot(top, label, feature)
    save_MAplot(top, label, feature)
}


get_juncs <- function(region){
    lt = load_counts(region, "junctions")
    load(lt[['count_file']])
    rse_df = rse_jxn
    genes = rowData(rse_df) %>% as.data.frame
    return(rownames(genes))
}
```

1.2.2 Cached functions

[3]:
```
get_mds <- function(){
    mds_file = "/ceph/projects/v4_phase3_paper/inputs/genotypes/mds/_m/
 ↪LIBD_Brain_TopMed.mds"
    mds = data.table::fread(mds_file) %>%
        rename_at(.vars = vars(starts_with("C")),
                  function(x){sub("C", "snpPC", x)}) %>%
        mutate_if(is.character, as.factor)
    return(mds)
}
```

```
memMDS <- memoise::memoise(get_mds)

get_overlapping_juncs <- function(){
    common_genes = intersect(intersect(get_juncs("caudate"),␣
↪get_juncs("dlpfc")), get_juncs("hippocampus"))
    return(common_genes)
}

memJuncs <- memoise(get_overlapping_juncs)

prep_data <- function(feature){
    datalist1 = list()
    datalist2 = list()
    for(region in c('dlpfc', 'caudate', 'hippocampus')){
        print(region)
        flush.console()
        lt = load_counts(region, feature)
        load(lt[['count_file']])
        tx_file = lt[['tx_file']]
        if (exists("rse_gene")){
            rse_df = rse_gene
        } else if (exists("rse_tx")){
            if(region == 'caudate' && exists("geneid")){
                counts <- fread(tx_file) %>% filter(gencodeTx %in% geneid) %>%
                    column_to_rownames("gencodeTx") %>%
                    select(-c(txLength, gencodeID, Symbol, gene_type)) %>%
                    select(colnames(rse_tx)) %>% as.matrix
                rse_df <- SummarizedExperiment(assays=SimpleList(counts=counts),
                                               rowData=annot,␣
↪colData=colData(rse_tx)) %>%
                    as("RangedSummarizedExperiment")
            } else {
                counts <- fread(tx_file) %>%␣
↪column_to_rownames("transcript_id") %>%
                    select(colnames(rse_tx)) %>% as.matrix
                annot <- fread(tx_file) %>% column_to_rownames("transcript_id")␣
↪%>%
                    select(-starts_with("R"))
                rse_df <- SummarizedExperiment(assays=SimpleList(counts=counts),
                                               rowData=annot,
                                               colData=colData(rse_tx)) %>%
                    as("RangedSummarizedExperiment")
                geneid = rownames(rowData(rse_df))
            }
        } else if (exists("rse_exon")){
            rse_df = rse_exon
```

```
        } else {
            rse_df = rse_jxn
        }
        keepIndex = which((rse_df$Dx %in% c("Control", "Schizo")) &
                          rse_df$Age > 17 & rse_df$Race %in% c("AA", "CAUC"))
        rse_df = rse_df[, keepIndex]
        rse_df$Sex <- factor(rse_df$Sex)
        rse_df <- jaffelab::merge_rse_metrics(rse_df)
        colData(rse_df)$RIN = sapply(colData(rse_df)$RIN,"[",1)
        colData(rse_df)$ERCCsumLogErr =␣
↪sapply(colData(rse_df)$ERCCsumLogErr,"[",1)
        rownames(colData(rse_df)) <- sapply(strsplit(rownames(colData(rse_df)),␣
↪"_"), "[", 1)
        pheno = colData(rse_df) %>% as.data.frame %>%
            inner_join(memMDS(), by=c("BrNum"="FID")) %>%
            distinct(RNum, .keep_all = TRUE)
        if(feature == "junctions"){
            datalist1[[region]] <- assays(rse_df)$counts[, pheno$RNum] %>% as.
↪data.frame %>%
                rownames_to_column("V1") %>% filter(V1 %in% memJuncs()) %>%
                column_to_rownames("V1") %>% t %>% as.data.frame
        }else{
            datalist1[[region]] <- assays(rse_df)$counts[, pheno$RNum] %>% t %>%
                as.data.frame
        }
        datalist2[[region]] <- pheno[, c('Sex', 'Dx', 'Age', 'rRNA_rate',␣
↪'totalAssignedGene',
                                         'RIN','ERCCsumLogErr',␣
↪'overallMapRate', 'mitoRate',
                                         'snpPC1', 'snpPC2', 'snpPC3',␣
↪'Region', 'BrNum', 'RNum')]
    }
    samples <- bind_rows(datalist2) %>% as.data.frame %>%
        mutate_if(is.numeric, rescale)
    subjects = samples$BrNum
    if(feature == "junctions"){
        genes = rowData(rse_df) %>% as.data.frame %>% rownames_to_column("V1")␣
↪%>%
            filter(V1 %in% memJuncs()) %>% column_to_rownames("V1")
    } else {
        genes = rowData(rse_df) %>% as.data.frame
    }
    counts <- bind_rows(datalist1) %>% t %>% as.data.frame
    colnames(counts) <- samples$RNum
    if(samples %>% has_rownames){
        samples = samples %>% select(-RNum)
```

```
    } else {
        samples = samples %>% column_to_rownames("RNum")
    }
    # Generate DGE list
    x <- DGEList(counts=counts[, row.names(samples)],
                 genes=genes, samples=samples)
    # Filter by expression
    design0 <- model.matrix(~Sex*Region, data=x$samples)
    keep.x <- filterByExpr(x, design=design0)
    x <- x[keep.x, , keep.lib.sizes=FALSE]
    print(paste('There are:', sum(keep.x), 'features left!', sep=' '))
    # Normalize library size
    x <- calcNormFactors(x, method="TMM")
    return(x)
}

memo_prepData <- memoise(prep_data)

plot_corrMatrix <- function(feature){
    x <- memo_prepData(feature)
    covs = c('Dx', 'Age', 'rRNA_rate', 'totalAssignedGene', 'RIN',
             'ERCCsumLogErr', 'overallMapRate', 'mitoRate', 'snpPC1',
             'snpPC2', 'snpPC3', 'Sex', 'Region', 'BrNum')
    form <- paste('~', glue_collapse(covs, sep=" + "))
    # Compute Canonical Correlation Analysis (CCA)
    # between all pairs of variables# returns absolute correlation value
    C = canCorPairs(form, x$samples)
    # Plot correlation matrix
    plotCorrMatrix( C )
}

memo_corrMatrix <- memoise(plot_corrMatrix)

get_voom <- function(feature){
    memo_corrMatrix(feature)
    x <- memo_prepData(feature)
    vobjDream = voomWithDreamWeights(x, get_model(),
                                     x$samples, plot=TRUE)
    return(vobjDream)
}

memo_voom <- memoise(get_voom)

fit_dream <- function(feature){
    x <- memo_prepData(feature)
    vobj <- memo_voom(feature)
    # define and then cbind contrasts
```

```
    L1 = getContrast(vobj, get_model(), x$samples,
                     c("RegionDLPFC", "RegionHIPPO"))
    L2 = getContrast(vobj, get_model(), x$samples,
                     c("SexM:RegionDLPFC", "SexM:RegionHIPPO"))
    # combine contrasts
    L = cbind(L1, L2)
    # Visualize contrasts
    plotContrasts(L)
    fit = dream(vobj, get_model(), x$samples, L=L)
    return(fit)
}

memo_dream <- memoise(fit_dream)

cal_res <- function(feature){
    ### Calculate residuals
    vobj <- memo_voom(feature)
    residList <- fitVarPartModel(vobj, get_null(), vobj$targets, fxn=residuals)
    do.call(rbind, residList) %>% as.data.frame %>%
        rownames_to_column %>% rename(Geneid=rowname) %>%
        fwrite(file=paste0(feature, '/residualized_expression.tsv'), sep='\t')
}

memo_res <- memoise(cal_res)
```

1.3 Main loop

[4]:
```
for(feature in c('genes', 'transcripts', 'junctions', 'exons')){
    flush.console()
    dir.create(feature)
    # Set parallel parameters
    print("Set parallel!")
    set_parallel(feature)
    # Prepare data
    print("Prepare data!")
    x <- memo_prepData(feature)
    save(x, file=paste0(feature, "/normData.RData"))
    # Preform voom
    print("Preform voom!")
    v <- memo_voom(feature)
    save(v, file=paste0(feature,'/voom_dream.RData'))
    # Fit model and apply eBayes
    print("Fit model!")
    fit = memo_dream(feature)
    # Save differential expression
    extract_de("SexM", "sex", fit, feature)
    extract_de("SexM:RegionDLPFC", "CvD_sex", fit, feature)
```

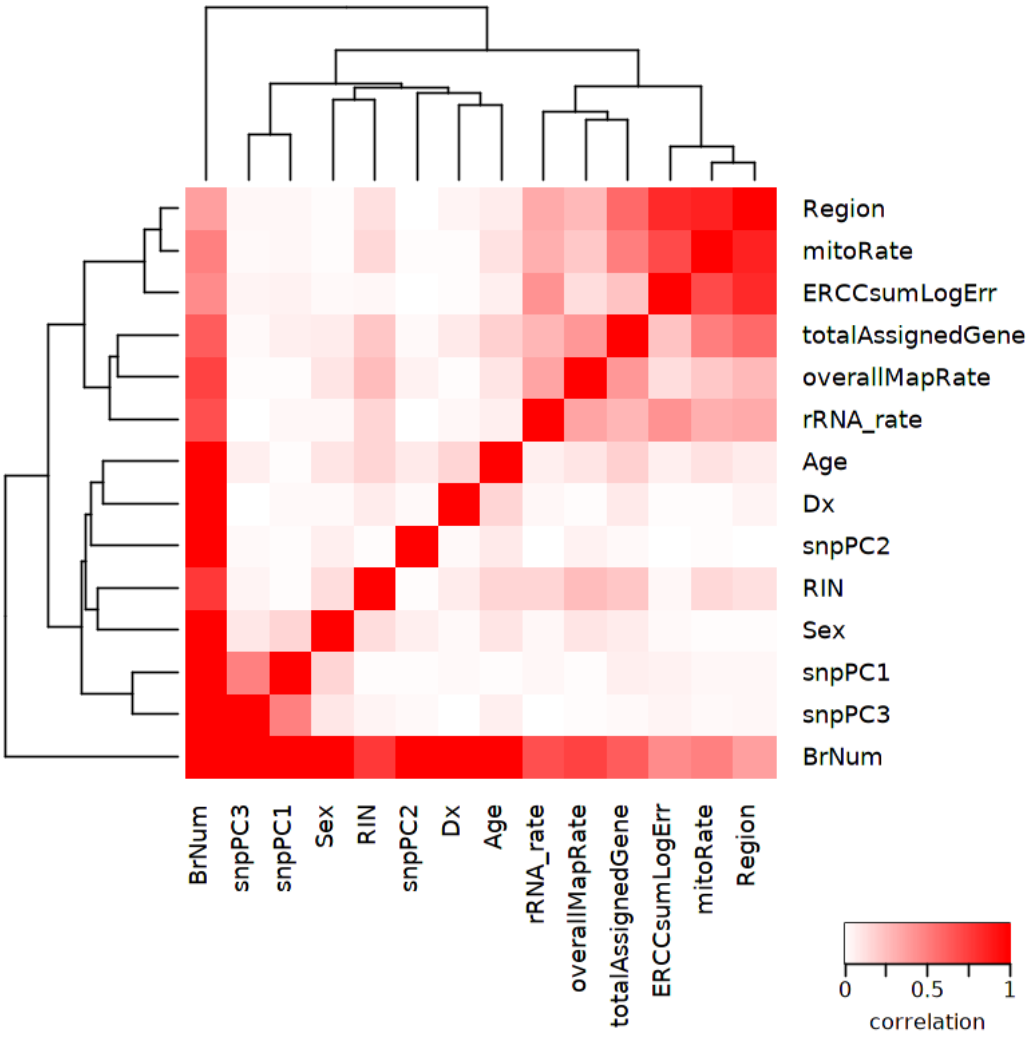
```
    extract_de("SexM:RegionHIPPO", "CvH_sex", fit, feature)
    extract_de("L2", "DvH_sex", fit, feature)
    # Calculate residuals
    print("Generated residualized expression!")
    memo_res(feature)
}
```

```
[1] "Set parallel!"
[1] "Prepare data!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 26385 features left!"
[1] "Preform voom!"
```

```
Warning message in canCorPairs(form, x$samples):
"Regression model may be problematic.
High colinearity between variables:
  Dx and BrNum
  Age and BrNum
  snpPC1 and BrNum
  snpPC2 and BrNum
  snpPC3 and BrNum
  Sex and BrNum"
Memory usage to store result: >2 Gb

Dividing work into 100 chunks…


Total:488 s


[1] "Fit model!"
```

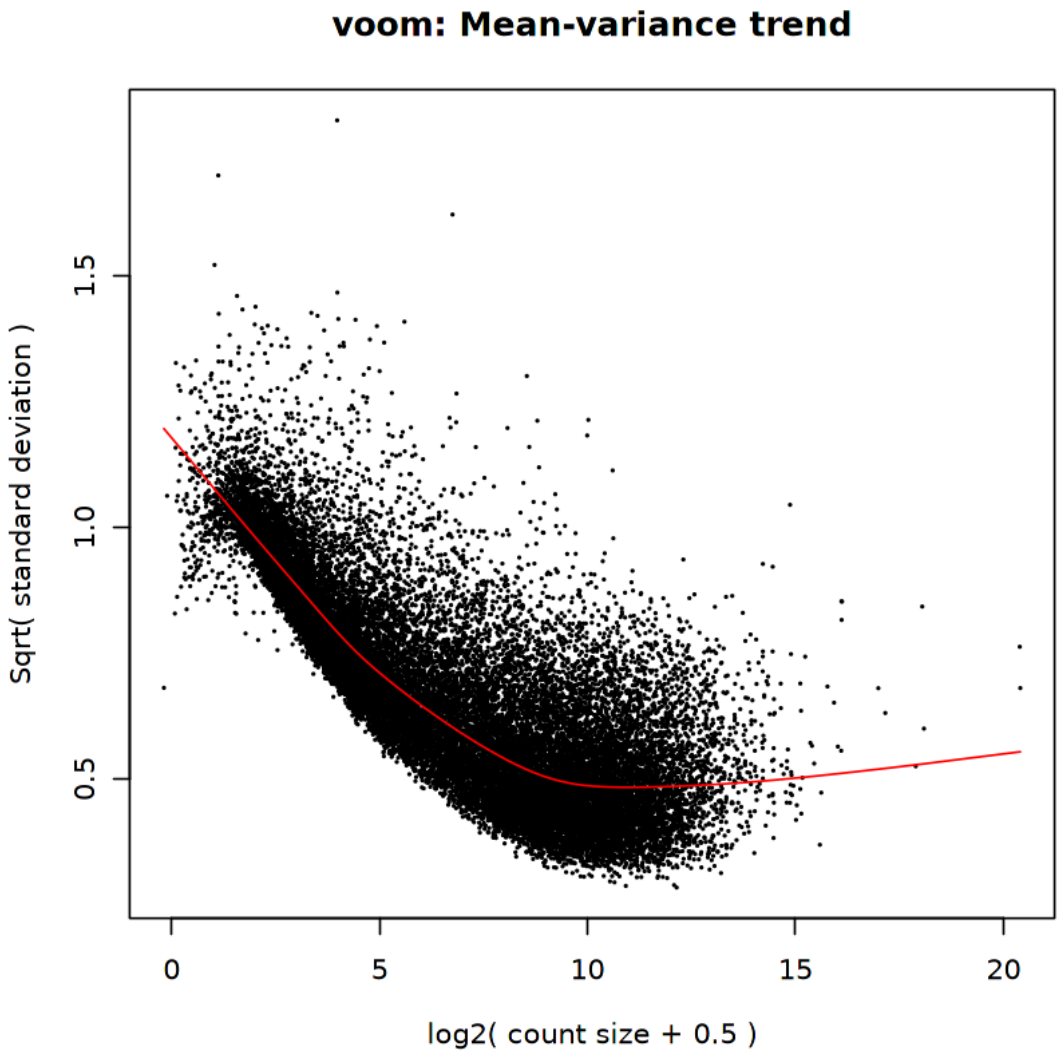
```
Dividing work into 100 chunks…


Total:772 s


[1] "Comparison for: sex"
[1] "There are: 704 DE features!"
[1] "Comparison for: CvD_sex"
[1] "There are: 668 DE features!"
[1] "Comparison for: CvH_sex"
[1] "There are: 23 DE features!"
[1] "Comparison for: DvH_sex"
[1] "There are: 26 DE features!"
[1] "Generated residualized expression!"
```

```
Memory usage to store result: >2 Gb

Dividing work into 100 chunks…


Total:403 s


[1] "Set parallel!"
[1] "Prepare data!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 118668 features left!"
[1] "Preform voom!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 118668 features left!"
Warning message in canCorPairs(form, x$samples):
"Regression model may be problematic.
High colinearity between variables:
  Dx and BrNum
  Age and BrNum
  snpPC1 and BrNum
  snpPC2 and BrNum
  snpPC3 and BrNum
  Sex and BrNum"
```




```
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 118668 features left!"
```

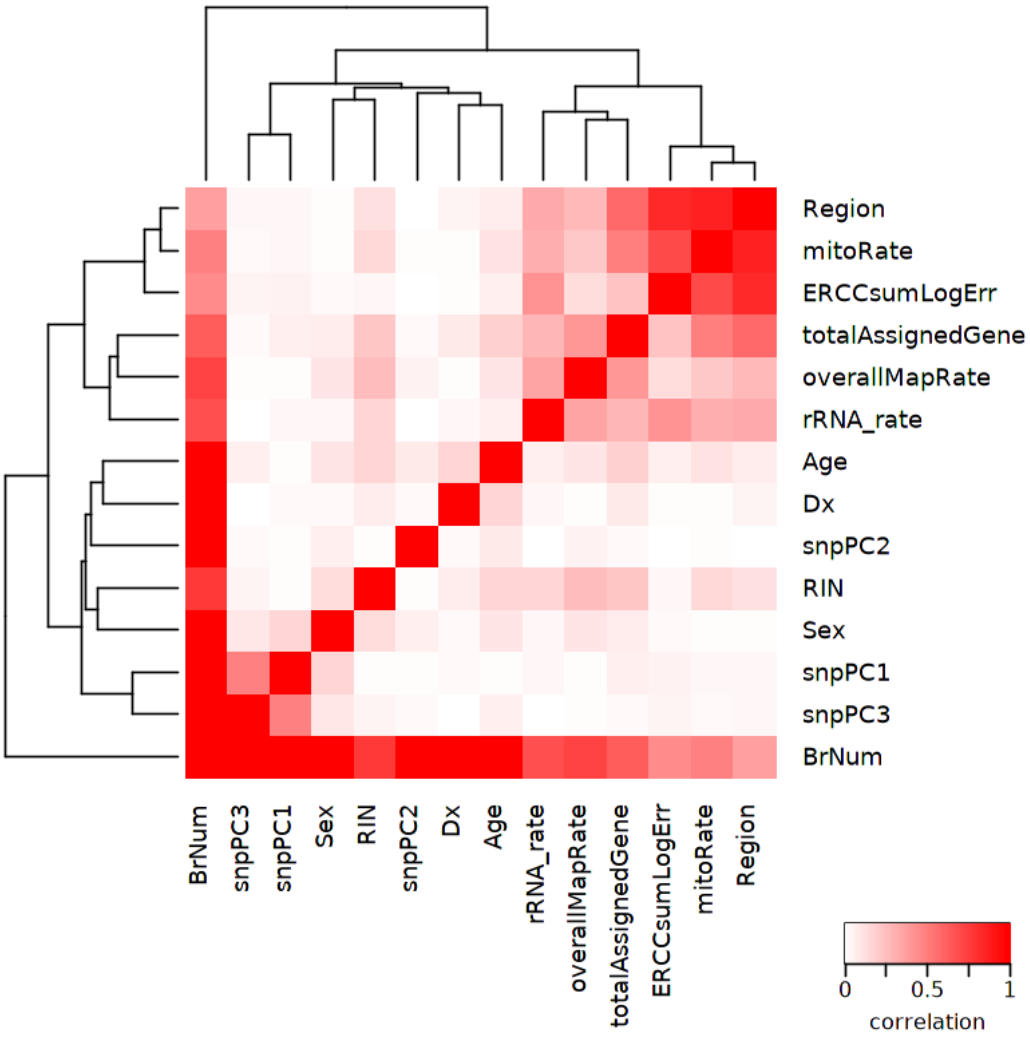
```
Memory usage to store result: >9 Gb

Dividing work into 100 chunks…


Total:1189 s
```




```
[1] "Fit model!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 118668 features left!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 118668 features left!"
```

```
Memory usage to store result: >9 Gb

Dividing work into 100 chunks…
```

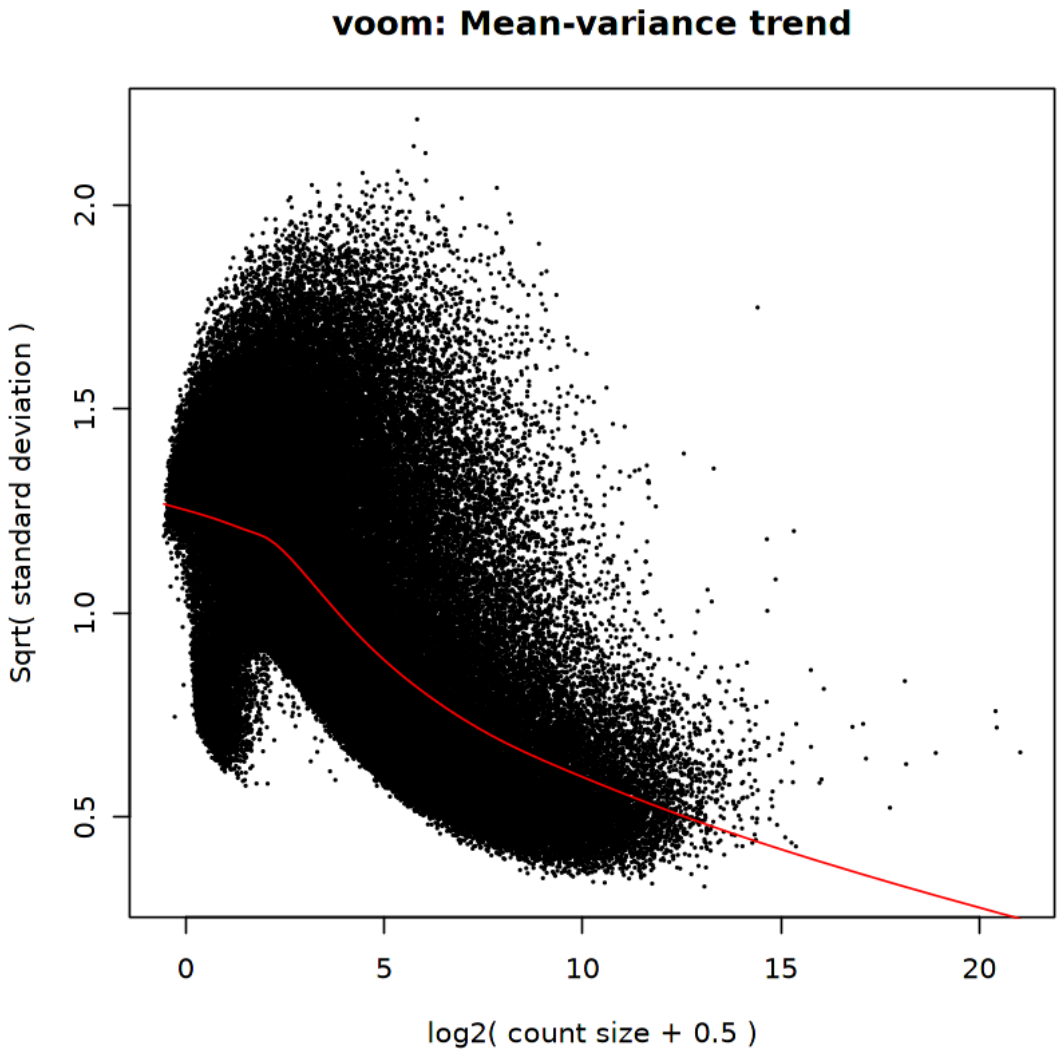
```
Total:1192 s
```




```
Dividing work into 100 chunks…

Total:2688 s

[1] "Comparison for: sex"
[1] "There are: 885 DE features!"
[1] "Comparison for: CvD_sex"
[1] "There are: 642 DE features!"
```

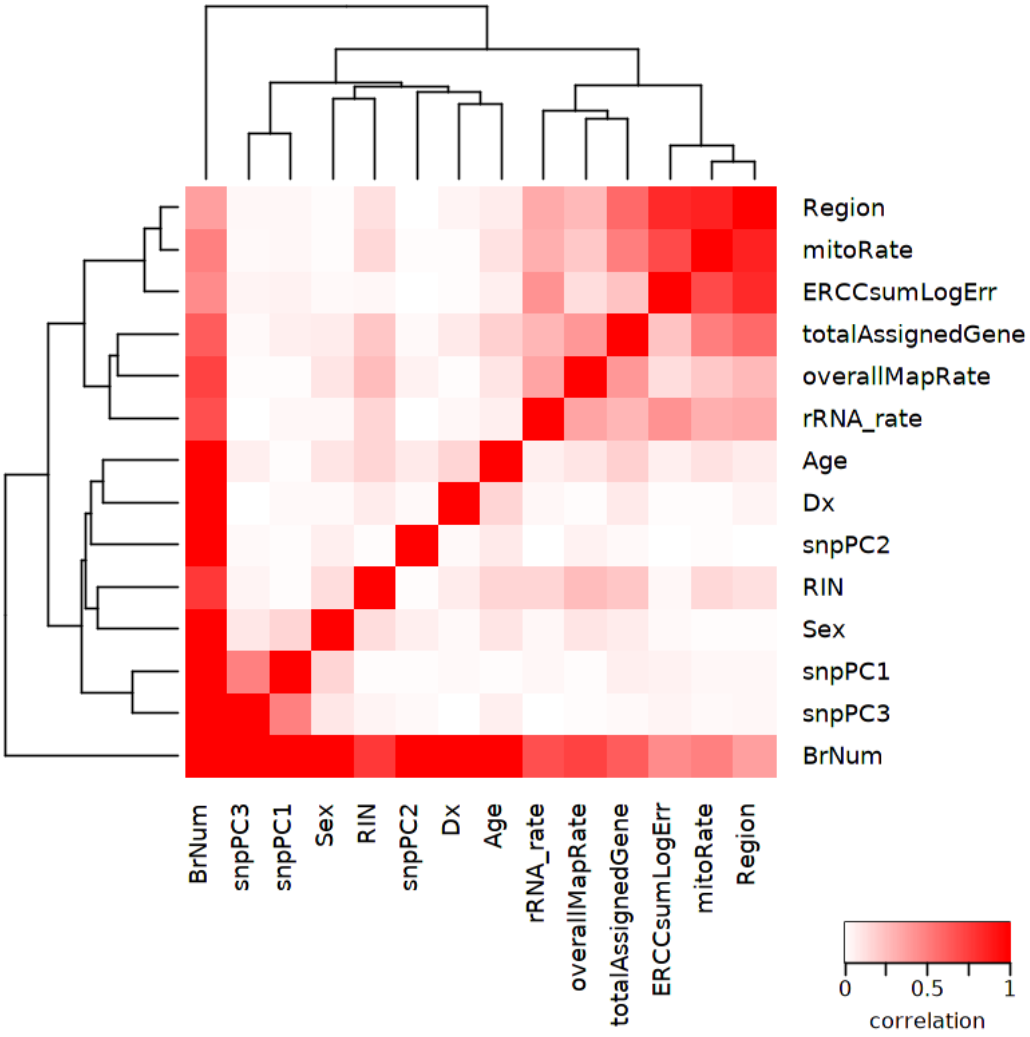
```
[1] "Comparison for: CvH_sex"
[1] "There are: 177 DE features!"
[1] "Comparison for: DvH_sex"
[1] "There are: 28 DE features!"
[1] "Generated residualized expression!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 118668 features left!"
Memory usage to store result: >9 Gb

Dividing work into 100 chunks…


Total:1184 s

Memory usage to store result: >9 Gb

Dividing work into 100 chunks…


Total:1163 s


[1] "Set parallel!"
[1] "Prepare data!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 173383 features left!"
[1] "Preform voom!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 173383 features left!"
Warning message in canCorPairs(form, x$samples):
"Regression model may be problematic.
High colinearity between variables:
  Dx and BrNum
  Age and BrNum
  snpPC1 and BrNum
  snpPC2 and BrNum
  snpPC3 and BrNum
  Sex and BrNum"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
```

```
[1] "There are: 173383 features left!"
Memory usage to store result: >13.2 Gb
Dividing work into 100 chunks…

Total:1709 s
```



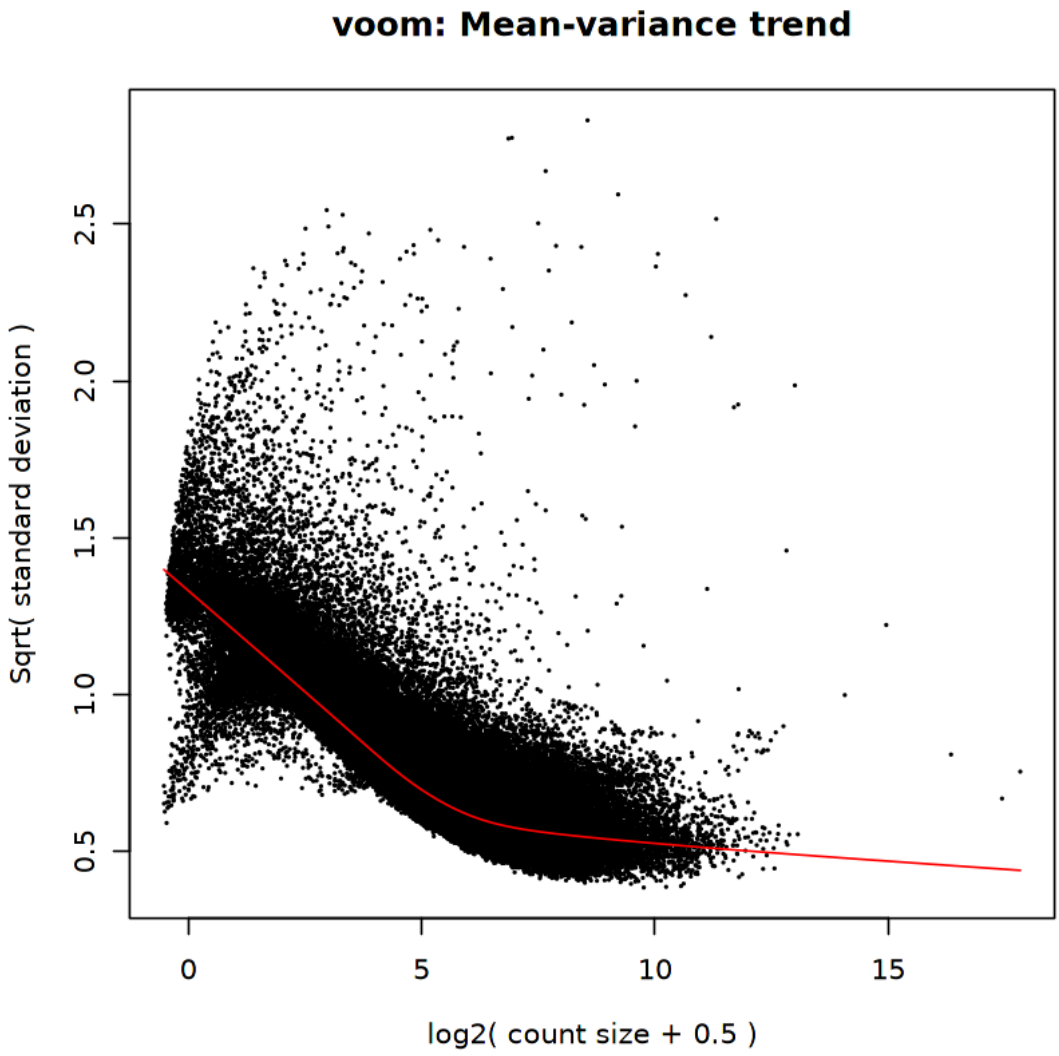

```
[1] "Fit model!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
```

```
[1] "There are: 173383 features left!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 173383 features left!"
Memory usage to store result: >13.2 Gb

Dividing work into 100 chunks…


Total:1724 s
```



```
Dividing work into 100 chunks…


Total:3978 s


[1] "Comparison for: sex"
[1] "There are: 1358 DE features!"
[1] "Comparison for: CvD_sex"
[1] "There are: 324 DE features!"
[1] "Comparison for: CvH_sex"
[1] "There are: 47 DE features!"
[1] "Comparison for: DvH_sex"
[1] "There are: 58 DE features!"
[1] "Generated residualized expression!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 173383 features left!"
Memory usage to store result: >13.2 Gb

Dividing work into 100 chunks…


Total:1741 s

Memory usage to store result: >13.1 Gb

Dividing work into 100 chunks…


Total:1714 s


[1] "Set parallel!"
[1] "Prepare data!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 382284 features left!"
[1] "Preform voom!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 382284 features left!"
Warning message in canCorPairs(form, x$samples):
"Regression model may be problematic.
High colinearity between variables:
```

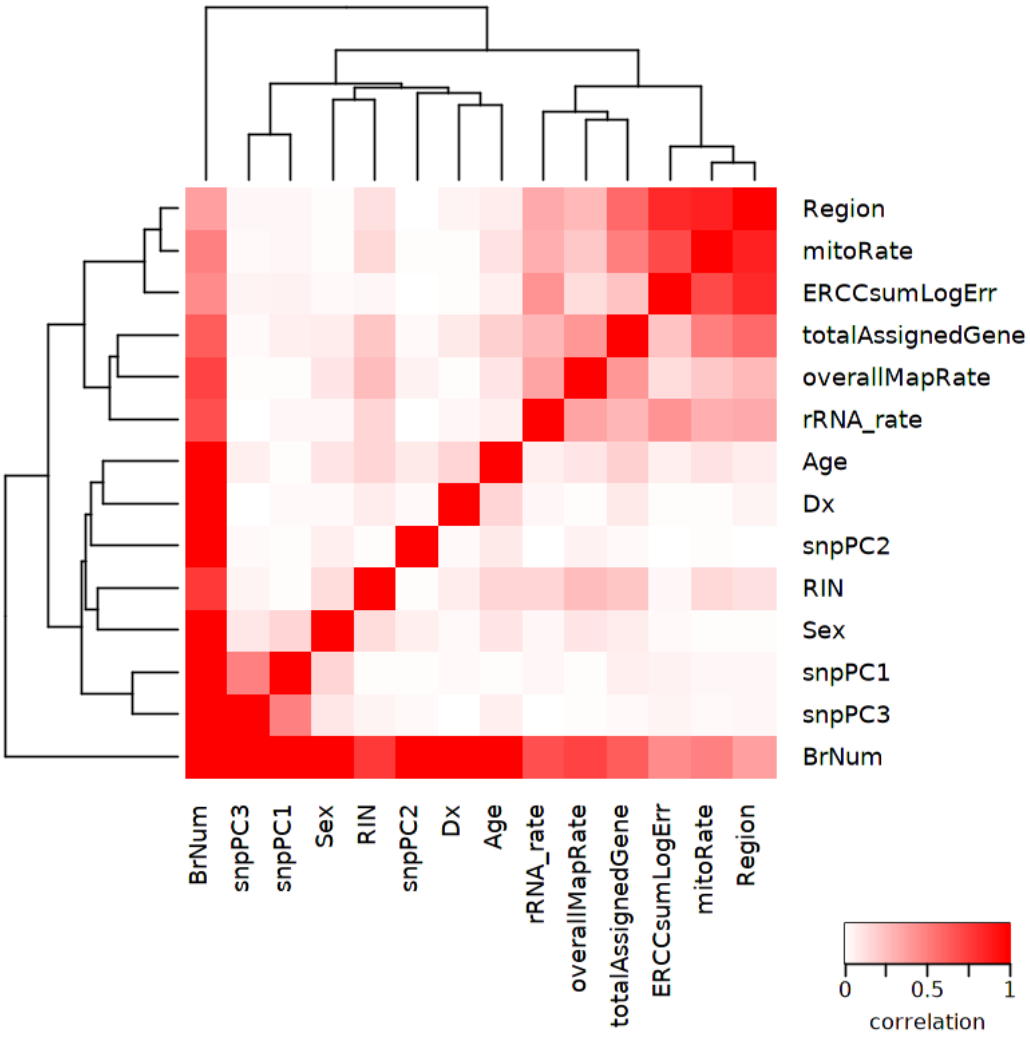
```
  Dx and BrNum
  Age and BrNum
  snpPC1 and BrNum
  snpPC2 and BrNum
  snpPC3 and BrNum
  Sex and BrNum"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 382284 features left!"
Memory usage to store result: >29.1 Gb

Dividing work into 100 chunks…


Total:3675 s
```




```
[1] "Fit model!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 382284 features left!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 382284 features left!"
```

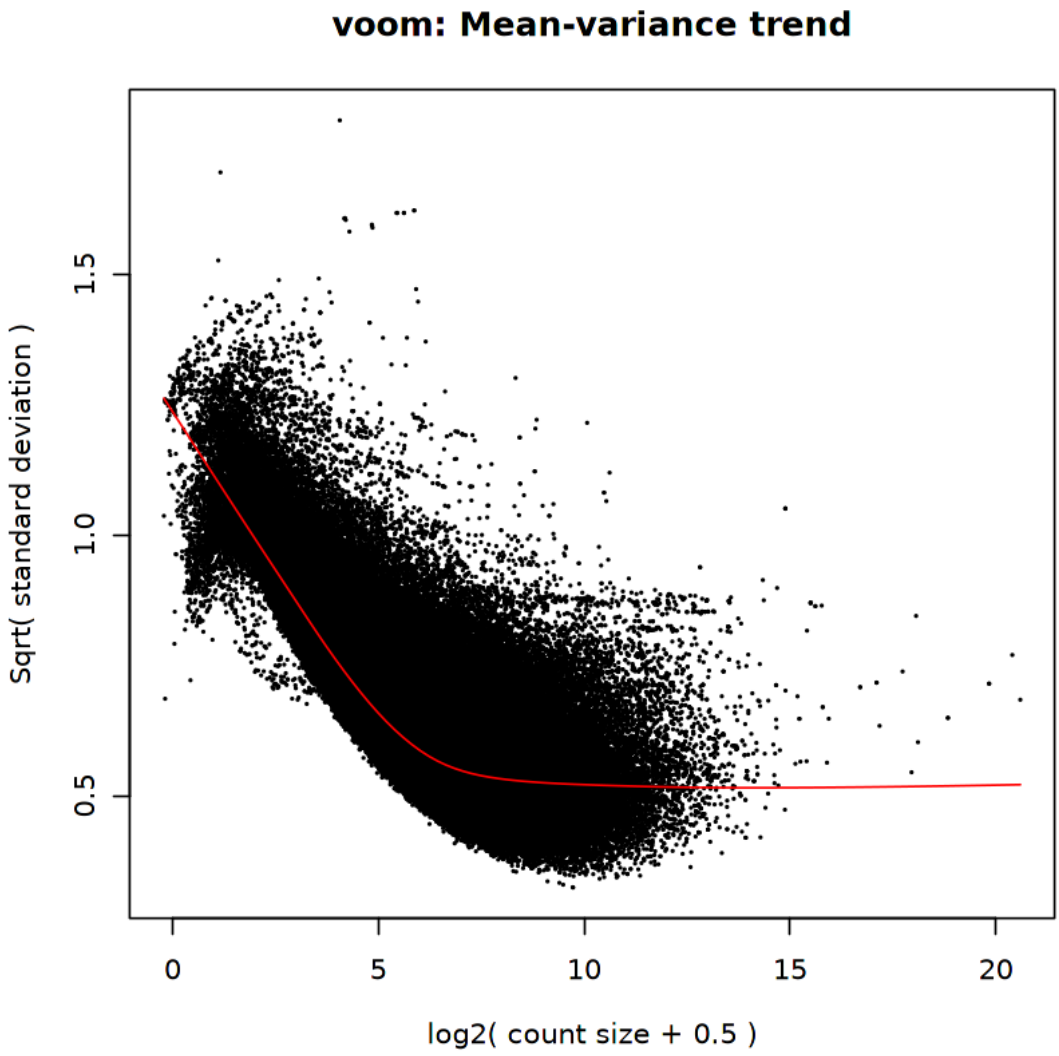
```
Memory usage to store result: >29.1 Gb

Dividing work into 100 chunks…
```

```
Total:3613 s
```




```
Dividing work into 100 chunks…

Total:%#4.0-1e s

[1] "Comparison for: sex"
[1] "There are: 4025 DE features!"
[1] "Comparison for: CvD_sex"
[1] "There are: 2254 DE features!"
```

```
[1] "Comparison for: CvH_sex"
[1] "There are: 129 DE features!"
[1] "Comparison for: DvH_sex"
[1] "There are: 131 DE features!"
[1] "Generated residualized expression!"
[1] "dlpfc"
[1] "caudate"
[1] "hippocampus"
[1] "There are: 382284 features left!"
Memory usage to store result: >29.1 Gb

Dividing work into 100 chunks…


Total:3642 s

Memory usage to store result: >29 Gb

Dividing work into 100 chunks…


Total:3604 s
```



1.4 Repreducibility Information

[5]:
```
Sys.time()
proc.time()
options(width = 120)
sessioninfo::session_info()
```

```
[1] "2021-08-04 05:59:53 EDT"
```

```
    user   system  elapsed
38648.65 11424.10 64363.82
```

```
 Session info
 setting  value
```

```
version  R version 4.0.3 (2020-10-10)
os       Arch Linux
system   x86_64, linux-gnu
ui       X11
language (EN)
collate  en_US.UTF-8
ctype    en_US.UTF-8
tz       America/New_York
date     2021-08-04

 Packages
package              * version    date       lib source
annotate               1.68.0     2020-10-27 [1] Bioconductor
AnnotationDbi          1.52.0     2020-10-27 [1] Bioconductor
assertthat             0.2.1      2019-03-21 [1] CRAN (R 4.0.2)
backports              1.2.1      2020-12-09 [1] CRAN (R 4.0.2)
base64enc              0.1-3      2015-07-28 [1] CRAN (R 4.0.2)
benchmarkme            1.0.7      2021-03-21 [1] CRAN (R 4.0.3)
benchmarkmeData        1.0.4      2020-04-23 [1] CRAN (R 4.0.3)
bigassertr             0.1.5      2021-07-08 [1] CRAN (R 4.0.3)
bigreadr               0.2.4      2021-04-10 [1] CRAN (R 4.0.3)
Biobase              * 2.50.0     2020-10-27 [1] Bioconductor
BiocGenerics         * 0.36.1     2021-04-16 [1] Bioconductor
BiocParallel         * 1.24.1     2020-11-06 [1] Bioconductor
bit                    4.0.4      2020-08-04 [1] CRAN (R 4.0.2)
bit64                  4.0.5      2020-08-30 [1] CRAN (R 4.0.2)
bitops                 1.0-7      2021-04-24 [1] CRAN (R 4.0.3)
blob                   1.2.1      2020-01-20 [1] CRAN (R 4.0.2)
boot                   1.3-25     2020-04-26 [2] CRAN (R 4.0.3)
broom                  0.7.8      2021-06-24 [1] CRAN (R 4.0.3)
cachem                 1.0.5      2021-05-15 [1] CRAN (R 4.0.3)
Cairo                  1.5-12.2   2020-07-07 [1] CRAN (R 4.0.2)
caTools                1.18.2     2021-03-28 [1] CRAN (R 4.0.3)
cellranger             1.1.0      2016-07-27 [1] CRAN (R 4.0.2)
cli                    3.0.0      2021-06-30 [1] CRAN (R 4.0.3)
codetools              0.2-16     2018-12-24 [2] CRAN (R 4.0.3)
colorRamps             2.3        2012-10-29 [1] CRAN (R 4.0.3)
colorspace             2.0-2      2021-06-24 [1] CRAN (R 4.0.3)
crayon                 1.4.1      2021-02-08 [1] CRAN (R 4.0.3)
data.table           * 1.14.0     2021-02-21 [1] CRAN (R 4.0.3)
DBI                    1.1.1      2021-01-15 [1] CRAN (R 4.0.2)
dbplyr                 2.1.1      2021-04-06 [1] CRAN (R 4.0.3)
DelayedArray           0.16.3     2021-03-24 [1] Bioconductor
digest                 0.6.27     2020-10-24 [1] CRAN (R 4.0.2)
disk.frame           * 0.5.0      2021-05-13 [1] CRAN (R 4.0.3)
doParallel             1.0.16     2020-10-16 [1] CRAN (R 4.0.3)
dplyr                * 1.0.7      2021-06-18 [1] CRAN (R 4.0.3)
edgeR                * 3.32.1     2021-01-14 [1] Bioconductor
```

```
ellipsis              0.3.2     2021-04-29 [1] CRAN (R 4.0.3)
evaluate              0.14      2019-05-28 [1] CRAN (R 4.0.2)
fansi                 0.5.0     2021-05-25 [1] CRAN (R 4.0.3)
farver                2.1.0     2021-02-28 [1] CRAN (R 4.0.3)
fastmap               1.1.0     2021-01-25 [1] CRAN (R 4.0.2)
forcats             * 0.5.1     2021-01-27 [1] CRAN (R 4.0.2)
foreach               1.5.1     2020-10-15 [1] CRAN (R 4.0.2)
fs                    1.5.0     2020-07-31 [1] CRAN (R 4.0.2)
fst                   0.9.4     2020-08-27 [1] CRAN (R 4.0.3)
future                1.21.0    2020-12-10 [1] CRAN (R 4.0.2)
future.apply          1.7.0     2021-01-04 [1] CRAN (R 4.0.2)
gargle                1.2.0     2021-07-02 [1] CRAN (R 4.0.3)
genefilter          * 1.72.1    2021-01-21 [1] Bioconductor
generics              0.1.0     2020-10-31 [1] CRAN (R 4.0.2)
GenomeInfoDb        * 1.26.7    2021-04-08 [1] Bioconductor
GenomeInfoDbData      1.2.4     2021-02-02 [1] Bioconductor
GenomicRanges       * 1.42.0    2020-10-27 [1] Bioconductor
ggplot2             * 3.3.5     2021-06-25 [1] CRAN (R 4.0.3)
globals               0.14.0    2020-11-22 [1] CRAN (R 4.0.2)
glue                * 1.4.2     2020-08-27 [1] CRAN (R 4.0.2)
googledrive           2.0.0     2021-07-08 [1] CRAN (R 4.0.3)
gplots                3.1.1     2020-11-28 [1] CRAN (R 4.0.2)
gtable                0.3.0     2019-03-25 [1] CRAN (R 4.0.2)
gtools                3.9.2     2021-06-06 [1] CRAN (R 4.0.3)
haven                 2.4.1     2021-04-23 [1] CRAN (R 4.0.3)
hms                   1.1.0     2021-05-17 [1] CRAN (R 4.0.3)
htmltools             0.5.1.1   2021-01-22 [1] CRAN (R 4.0.2)
httr                  1.4.2     2020-07-20 [1] CRAN (R 4.0.2)
IRanges             * 2.24.1    2020-12-12 [1] Bioconductor
IRdisplay             1.0       2021-01-20 [1] CRAN (R 4.0.2)
IRkernel              1.2       2021-05-11 [1] CRAN (R 4.0.3)
iterators             1.0.13    2020-10-15 [1] CRAN (R 4.0.2)
jaffelab              0.99.30   2021-02-02 [1] Github (LieberInstitute/
↪jaffelab@42637ff)
jsonlite              1.7.2     2020-12-09 [1] CRAN (R 4.0.2)
KernSmooth            2.23-17   2020-04-26 [2] CRAN (R 4.0.3)
lattice               0.20-41   2020-04-02 [2] CRAN (R 4.0.3)
lifecycle             1.0.0     2021-02-15 [1] CRAN (R 4.0.3)
limma               * 3.46.0    2020-10-27 [1] Bioconductor
listenv               0.8.0     2019-12-05 [1] CRAN (R 4.0.2)
lme4                  1.1-27.1  2021-06-22 [1] CRAN (R 4.0.3)
lmerTest              3.1-3     2020-10-23 [1] CRAN (R 4.0.3)
locfit                1.5-9.4   2020-03-25 [1] CRAN (R 4.0.2)
lubridate             1.7.10    2021-02-26 [1] CRAN (R 4.0.3)
magrittr              2.0.1     2020-11-17 [1] CRAN (R 4.0.2)
MASS                  7.3-53    2020-09-09 [2] CRAN (R 4.0.3)
Matrix                1.3-4     2021-06-01 [1] CRAN (R 4.0.3)
```

```
MatrixGenerics       * 1.2.1      2021-01-30 [1] Bioconductor
matrixStats          * 0.59.0     2021-06-01 [1] CRAN (R 4.0.3)
memoise              * 2.0.0      2021-01-26 [1] CRAN (R 4.0.2)
mgcv                 * 1.8-33     2020-08-27 [2] CRAN (R 4.0.3)
minqa                  1.2.4      2014-10-09 [1] CRAN (R 4.0.2)
modelr                 0.1.8      2020-05-19 [1] CRAN (R 4.0.2)
munsell                0.5.0      2018-06-12 [1] CRAN (R 4.0.2)
nlme                 * 3.1-152    2021-02-04 [1] CRAN (R 4.0.3)
nloptr                 1.2.2.2    2020-07-02 [1] CRAN (R 4.0.2)
numDeriv               2016.8-1.1 2019-06-06 [1] CRAN (R 4.0.2)
parallelly             1.26.1     2021-06-30 [1] CRAN (R 4.0.3)
pbdZMQ                 0.3-5      2021-02-10 [1] CRAN (R 4.0.3)
pbkrtest               0.5.1      2021-03-09 [1] CRAN (R 4.0.3)
pillar                 1.6.1      2021-05-16 [1] CRAN (R 4.0.3)
pkgconfig              2.0.3      2019-09-22 [1] CRAN (R 4.0.2)
plyr                   1.8.6      2020-03-03 [1] CRAN (R 4.0.2)
prettyunits            1.1.1      2020-01-24 [1] CRAN (R 4.0.2)
progress               1.2.2      2019-05-16 [1] CRAN (R 4.0.2)
pryr                   0.1.4      2018-02-18 [1] CRAN (R 4.0.3)
purrr                * 0.3.4      2020-04-17 [1] CRAN (R 4.0.2)
R6                     2.5.0      2020-10-28 [1] CRAN (R 4.0.2)
rafalib                1.0.2      2021-04-26 [1] Github (ririzarr/
↪rafalib@2580666)
RColorBrewer           1.1-2      2014-12-07 [1] CRAN (R 4.0.2)
Rcpp                   1.0.7      2021-07-07 [1] CRAN (R 4.0.3)
RCurl                  1.98-1.3   2021-03-16 [1] CRAN (R 4.0.3)
readr                * 1.4.0      2020-10-05 [1] CRAN (R 4.0.2)
readxl                 1.3.1      2019-03-13 [1] CRAN (R 4.0.2)
repr                   1.1.3      2021-01-21 [1] CRAN (R 4.0.2)
reprex                 2.0.0      2021-04-02 [1] CRAN (R 4.0.3)
reshape2               1.4.4      2020-04-09 [1] CRAN (R 4.0.2)
rlang                  0.4.11     2021-04-30 [1] CRAN (R 4.0.3)
RSQLite                2.2.7      2021-04-22 [1] CRAN (R 4.0.3)
rstudioapi             0.13       2020-11-12 [1] CRAN (R 4.0.2)
rvest                  1.0.0      2021-03-09 [1] CRAN (R 4.0.3)
S4Vectors            * 0.28.1     2020-12-09 [1] Bioconductor
scales               * 1.1.1      2020-05-11 [1] CRAN (R 4.0.2)
segmented              1.3-4      2021-04-22 [1] CRAN (R 4.0.3)
sessioninfo            1.1.1      2018-11-05 [1] CRAN (R 4.0.2)
snow                   0.4-3      2018-09-14 [1] CRAN (R 4.0.2)
stringi                1.7.3      2021-07-16 [1] CRAN (R 4.0.3)
stringr              * 1.4.0      2019-02-10 [1] CRAN (R 4.0.2)
SummarizedExperiment * 1.20.0     2020-10-27 [1] Bioconductor
survival               3.2-7      2020-09-28 [2] CRAN (R 4.0.3)
sva                  * 3.38.0     2020-10-27 [1] Bioconductor
tibble               * 3.1.2      2021-05-16 [1] CRAN (R 4.0.3)
tidyr                * 1.1.3      2021-03-03 [1] CRAN (R 4.0.3)
```

```
 tidyselect             1.1.1      2021-04-30 [1] CRAN (R 4.0.3)
 tidyverse            * 1.3.1      2021-04-15 [1] CRAN (R 4.0.3)
 utf8                   1.2.1      2021-03-12 [1] CRAN (R 4.0.3)
 uuid                   0.1-4      2020-02-26 [1] CRAN (R 4.0.2)
 variancePartition    * 1.20.0     2020-10-27 [1] Bioconductor
 vctrs                  0.3.8      2021-04-29 [1] CRAN (R 4.0.3)
 withr                  2.4.2      2021-04-18 [1] CRAN (R 4.0.3)
 XML                    3.99-0.6   2021-03-16 [1] CRAN (R 4.0.3)
 xml2                   1.3.2      2020-04-23 [1] CRAN (R 4.0.2)
 xtable                 1.8-4      2019-04-21 [1] CRAN (R 4.0.2)
 XVector                0.30.0     2020-10-27 [1] Bioconductor
 zlibbioc               1.36.0     2020-10-27 [1] Bioconductor

[1] /home/jbenja13/R/x86_64-pc-linux-gnu-library/4.0
[2] /usr/lib/R/library
```