RNA editing pipeline

Step One: de novo calling with Jacusa2

manual github

```
ml jacusa2

java -jar $JACUSA2_JAR call-1 -r file.out /full/path/to/RAPiD/sample/file
.bam -p 10 -a D,M,Y,E:file=/sc/arion/projects/ad-omics/data/references/ed
iting/hg38-blacklist.v2_sort.bed:type=BED -s -m 20 -R /sc/arion/projects/
ad-omics/data/references/GRCh38_references/GRCh38.primary_assembly.genome
.fa -P FR-SECONDSTRAND -F 1024

sed -i '1d' file.out
```

Some notes about the Jacusa output format

Column 5 is the test-statistic of the likelihood ratio test for comparing two conditions I and II. Base call count vectors D-I and D-II are modelled with the Dirichlet-Multinomial distribution see manual 5.1.3.

Higher values of the test-statistic indicate a higher divergence of base call vectors between conditions.

When only one condition is provided, the command call-1 creates an *in silico* condition by using available reference information to replace non-reference base calls, creating synthetic base call vectors and applying the likelihood ratio test defined above.

Column 7 is the bases, where read counts for each base is provided in the following vector format: A, C, G, T.

Jacusa flags explained

- -m flag filters positions with MAPQ < MIN-MAPQ for all conditions (default: 20)
- -P flag is for library type (first strand, second strand, or unstranded)
- -a flag is the artifact/feature filter

    - D: filters potential false positive variants adjacent to indels, adjacent to read start/end (6bp), adjacent to splice sites (6bp)
    - M: max allowed alleles per site (default 2)
    - Y: filter wrong variant calls within homopolymers (dafault length 7)
    - E: exclude sites contained in file

- -p flag specifies threads
- -s flag stores feature-filtered results in another file (= RESULT-FILE.filtered if no argument) or (= FILTERED-FILE)
- -F 1024 removes PCR duplicates

Step Two: filtering and aggregating across samples to make coverage and editing ratio matrices

In order to pass the filtering thresholds at this stage, sites must be:

- total read coverage >=10
- edited read coverage >=2
- drop sites where ref=alt or alt count is 0
- drop multiallelic sites

```
library(splitstackshape)
library(dplyr)
library(tidyverse)
library(plyr)

#restructuring Jacusa2 output and light sample-level filtering
setwd("/path/to/Jacusa2/outputs/")
extension <- "out"
fileNames <- Sys.glob(paste("*.", extension, sep = ""))
fileNumbers <- seq(fileNames)

for (fileNumber in fileNumbers){
  out <- read.delim(fileNames[fileNumber], header = T, sep = "\t") #read in
  OUT <- cSplit(out, "bases11",",", direction = "wide") #split base vector
  OUT <- OUT[,-c(2,4,6,7,8)] #drop "start" because we only need "end" posit
  colnames(OUT) <- c("chr", "pos", "score", "ref", "A", "C", "G", "T") #ass
  OUT$totcov <- rowSums(OUT[,c(5:8)]) #creating total coverage column for e
  DF <- OUT %>% mutate(refcov = case_when(ref == "T" ~ .[[8]], ref == "A" ~
  DF <- DF[,-c(1:2)] #dropping chr & pos separated
  DF <- DF[,c(9,2,1,7,8,3,4,5,6)] #tidying column order
  DFnew <- DF %>% pivot_longer(!c(chrpos, ref, score, refcov, totcov), name
  covThreshold <- 10
```

```
  DFnew <- DFnew[which(DFnew$ref != DFnew$alt & DFnew$totcov >= covThreshol
  DFnewer <- DFnew[!duplicated(DFnew$chrpos),] #drop multiallelic sites
  df <- DFnewer %>% mutate(ESid = paste0(chrpos, ":", ref, ":", alt)) #crea
  colnames(df) <- c("chrpos", "ref", "score", "totcov", "refcov", "alt", fi
  colnames(df) <- gsub(pattern = ".out", replacement = "", colnames(df))
  temp <- df[,c(8,2,5,6,7,4,3)]
  write.table(temp, file = paste0(fileNames[fileNumber], ".Jacusa2filt"), q
}

#aggregating across samples
files <- dir(path = "/path/to/Jacusa2/outputs/", pattern = "*.Jacusa2filt
")
data <- files %>% map(read_tsv) #%>% join_all(,by = "ESid", type = "full"
)
aggDF <- reduce(data, full_join, by = "ESid")
covMat <- map(data, ~{
  d <- select(.x, ESid, totcov)
  colnames(d)[2] <- colnames(.x)[5]
  return(d)
}) %>% reduce(full_join, by = "ESid")

ratioMat <- map(data, ~{
  .x$edrat <- .x[,5]/.x[,6]
  d <- select(.x, ESid, edrat)
  colnames(d)[2] <- colnames(.x)[5]
  return(d)
}) %>% reduce(full_join, by = "ESid")

#cohort level-filtering: editing sites must validate across 50% of sample
s and have editing efficiency of at least 10%
N <- ceiling(length(fileNumbers) * 0.5) #set threshold for inter-donor va
lidation
covMat <- covMat[which(rowSums(!is.na(covMat[,-1])) >= N),]
ratioMat <- ratioMat[which(rowMeans(ratioMat[,-1]) >= 0.1),] #set editing
 ratio threshold

covMatfinal <- subset(covMat, (covMat$ESid %in% intersect(covMat$ESid, ra
tioMat$ESid)))
ratioMatfinal <- subset(ratioMat, (ratioMat$ESid %in% intersect(covMat$ES
id, ratioMat$ESid)))

save(covMatfinal, file = "covMat.Rda")
save(ratioMatfinal, file = "ratioMat.Rda")
```

Step Three: format editing sites from step two into Annovar input

```
library(dplyr)

load("ratioMat.Rda")
load("covMat.Rda")

annovar <- data.frame(ratioMat$ESid, do.call(rbind, strsplit(ratioMat$ESi
d, split = ":", fixed = TRUE)))
annovar <- annovar[,c(2,3,3,4,5)]
colnames(annovar) <- c("Chr", "Start", "Stop", "Ref", "Alt")

write.table(annovar, file = "file.avinput",
            append = FALSE, quote = FALSE, sep = "\t",
            row.names = FALSE, col.names = TRUE)
```

Step Four: Annovar annotation of common SNPs, repeat elements, genes, and gene regions

```
ml annovar
ml bcftools

IN="/full/path/to/file.avinput"
OUT="/full/path/to/file.myanno"

table_annovar.pl $IN /sc/arion/projects/ad-omics/data/references/editing/
humandb/ -buildver hg38 -out $OUT -remove -protocol refGene,dbsnp153Commo
nSNV,gnomad30_genome,phastConsElements30way,rmsk,rediportal_012920 -opera
tion g,f,f,r,r,f --argument ,,,'--colsWanted 5','--colsWanted 10&11&12',
-nastring "." --otherinfo --thread 10 --maxGeneThread 10

awk 'BEGIN{OFS=FS="\t"}{if ( ($11=="." || $11=="dbSNP153CommonSNV") && (
$12<0.05 || $12=="AF")) print $0}' file.myanno.hg38_multianno.txt > file.
myanno.hg38_multianno.txt.noCommon.txt
```

Step Five: Filter coverage matrix, editing ratio matrix using annotations

```
annoOut <- read.delim("file.myanno.hg38_multianno.txt.noCommon.txt", sep
= "\t", header = T)
anno <- annoOut[,c(1:7,26)]
colnames(anno) <- c("Chr", "Start", "Stop", "Ref", "Alt", "Region", "Gene
", "Repeat")
anno <- anno %>% mutate(ESid = paste0(Chr, ":", Start, ":", Ref, ":", Alt
))
ESannotated <- anno[,c(9,6,7,8)]

load("ratioMat.Rda")
load("covMat.Rda")
coverageMatrixFinal <- subset(covMatfinal, (covMatfinal$ESid %in% ESannot
ated$ESid))
ratioMatrixFinal <- subset(ratioMatfinal, (ratioMatfinal$ESid %in% ESanno
tated$ESid))
ESannotatedFinal <- subset(ESannotated, (ESannotated$ESid %in% intersect(
ratioMatrixFinal$ESid, coverageMatrixFinal$ESid)))
save(coverageMatrixFinal, file = "FinalCoverageMatrix.Rda")
save(ratioMatrixFinal, file = "FinalRatioMatrix.Rda")
save(ESannotatedFinal, file = "FinalAnnotationsMatrix.Rda")
```