Step1

Modify the meta.data.df variable.
Write the replicate names (i.e. rep1, rep2, IVT) and write the bam.name of the bam files relative to the replicates you want to analize. (Just names of bam files, not paths).
The replicate names will refer to folders in which the results for each experiment will be saved.

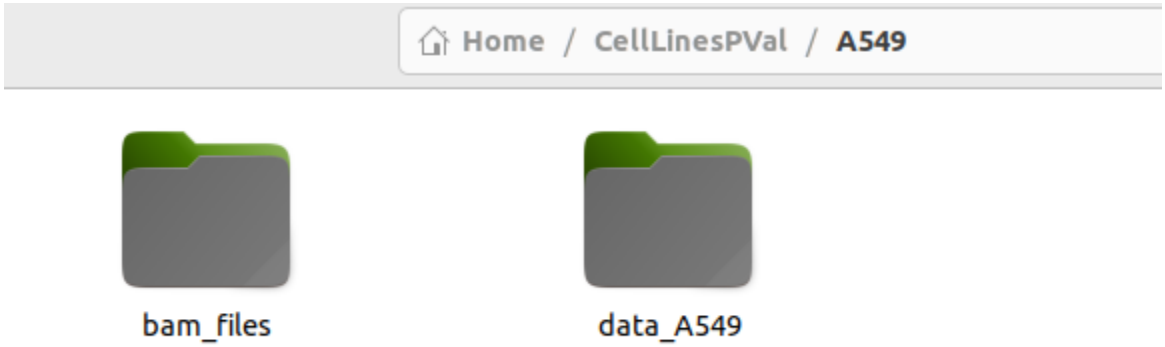
```
library(Rsamtools)
library(rtracklayer)
library(BSgenome)

print('step1')

############ INPUTS ############
################################################################################################ modify

#meta.data.df = data.frame("replicate.name"=c("PB1","PB2","Undiff1","Undiff2","Undiff3","IVT"),
#                          "bam.name" = c("SH-SY5Y_PB_Undiff_Direct_rep1.bam",
#                                         "SH-SY5Y_PB_Undiff_Direct_rep2.bam",
#                                         "SH-SY5Y_Undiff_Direct_rep1.bam",
#                                         "SH-SY5Y_Undiff_Direct_rep2.bam",
#                                         "SH-SY5Y_Undiff_Direct_rep3.bam",
#                                         "SH-SY5Y_Diff_IVT_rep1.bam"))


# meta.data.df = data.frame("replicate.name"=c("rep1","rep2","IVT"),
#                           "bam.name" = c("SH-SY5Y_undiff_Direct_rep1.hg38v10.bam", "SH-SY5Y_undiff_Direct_rep2.hg38v10.bam

meta.data.df = data.frame("replicate.name"=c("rep3"),
                          "bam.name" = c("SH-SY5Y_undiff_Direct_rep3.hg38v10.bam"))

hg38.fa = FaFile("~/CellLinesPVal/GRCh38.p10.genome.fa")
g = readGFF("~/CellLinesPVal/gencode/gencode.v27.annotation.gff3")
```

Then provide the path to the bam files and the path to the output folder.
The output folder can be located where desired, but ih has to be named init_gene_pileup and created BEFORE running the step1.

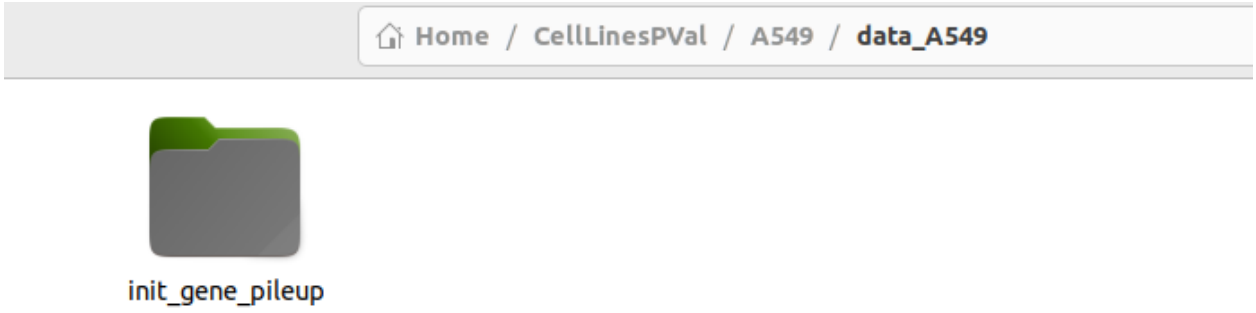
```
63  acceptable.bases = c("T", "A", "G", "+", "-", "C")
64
65  for (k in c(1:nrow(meta.data.df))){
66
67
68    ################################################################################################ modify
69    ####################create init_gene_pileup folder and all the replicate.name folders#####################
70    bam.dir = paste0("~/CellLinesPVal/SH-SY5Y/bam_files/",meta.data.df$bam.name[k])
71    #out.dir = paste0("~/repository/2023/Nanopore_workflow/data/init_gene_pileup/",meta.data.df$replicate.name[k],"/")
72    out.dir = paste0("~/CellLinesPVal/SH-SY5Y/data_SH-SY5Y/init_gene_pileup/",meta.data.df$replicate.name[k])
73
```

Example:

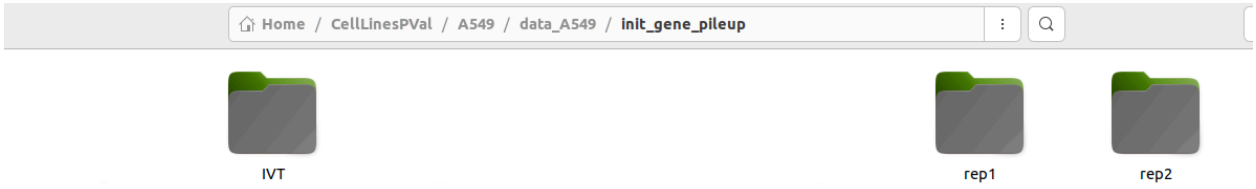
My bam folder is under /Home/CellLinesPVal/A549/bam_files
bam.dir = paste0("/Home/CellLinesPVal/A549/bam_files",meta.data.df$bam.name[k])



And I want the output to be saved in the folder data_A549. So before running step1 I'll create a folder named init_gene_pileup inside data_A549



And Then, inside init_gene_pileup i'll create as many folders as the replicates specified at the beginning.



Run the script (source)

Step2

Specify data.dir= the path to init_gene_pileup folder you created in step1.

```
rm(list = ls());cat("\014")
#CREATE A FOLDER NAMED Merged IN THE init_gene_pileup FOLDER BEFORE RUNNING EVERYTHING, MODIFY ROW 4 AND THE DATA FRAME IN LI
# Find the list of genes for which there is at least 1 pileup file
data.dir = "~/CellLinesPVal/SH-SY5Y/data_SH-SY5Y/init_gene_pileup/"

replicates = list.files(data.dir)
replicates = replicates[-which(replicates == "Merged")]
target.genes  = c()
```

Modify the merge.df accordingly to the replicates you analized in step1.
The same fields will be written for all the replicates, which are chr, position, strand, ttarget nucleotide, N_reads, A, T,C,G,del,ins. So add a convenient suffix to distinguish the replicates one from the others.

```
    # if there are any covered positions
    if (length(positions)>0){
      merged.df = data.frame("chr"=gene.chr, "position"=positions, "strand"=gene.strand,"target.nucleotide"="",
                             "N_reads_rep1"=0, "A_rep1"=0, "T_rep1"=0, "C_rep1"=0, "G_rep1"=0 ,"del_rep1"=0, "ins_rep1"=0,
                             "N_reads_rep2"=0, "A_rep2"=0, "T_rep2"=0, "C_rep2"=0, "G_rep2"=0 ,"del_rep2"=0, "ins_rep2"=0,
                             "N_reads_rep3"=0, "A_rep3"=0, "T_rep3"=0, "C_rep3"=0, "G_rep3"=0 ,"del_rep3"=0, "ins_rep3"=0,
                             "N_reads_IVT"=0, "A_IVT"=0, "T_IVT"=0, "C_IVT"=0, "G_IVT"=0 ,"del_IVT"=0, "ins_IVT"=0)

```

Then create a folder named Merged under the init_gene_pileup folder

Run the script (source)

Step3

Specify the data.dir=path to the merged folder created in step3
Specify the output directory out.dir=path to a folder under init_gene_pileup named Processd_pileup.

```
######################################## path to merged raw pileups for each gene
data.dir = "~/CellLinesPVal/SH-SY5Y/data_SH-SY5Y/init_gene_pileup/Merged/"

# path to save the processed data
######################## YOU NEED TO CREATE THIS FOLDER BEFORE RUNNING THE SCRIPT
out.dir = "~/CellLinesPVal/SH-SY5Y/data_SH-SY5Y/init_gene_pileup/Processed_pileup/"
```

Modify the raw.pieleup.df.2 variable according to the names you choosed for your replicates

```
53
54      # correct pileup values too
55      raw.pileup.df.2 = raw.pileup.df
56
57  ###############################################################################
58
59      raw.pileup.df.2$A_rep1 = raw.pileup.df$T_rep1
60      raw.pileup.df.2$T_rep1 = raw.pileup.df$A_rep1
61      raw.pileup.df.2$C_rep1 = raw.pileup.df$G_rep1
62      raw.pileup.df.2$G_rep1 = raw.pileup.df$C_rep1
63
64      raw.pileup.df.2$A_rep2 = raw.pileup.df$T_rep2
65      raw.pileup.df.2$T_rep2 = raw.pileup.df$A_rep2
66      raw.pileup.df.2$C_rep2 = raw.pileup.df$G_rep2
67      raw.pileup.df.2$G_rep2 = raw.pileup.df$C_rep2
68
69      raw.pileup.df.2$A_rep3 = raw.pileup.df$T_rep3
70      raw.pileup.df.2$T_rep3 = raw.pileup.df$A_rep3
71      raw.pileup.df.2$C_rep3 = raw.pileup.df$G_rep3
72      raw.pileup.df.2$G_rep3 = raw.pileup.df$C_rep3
73
98
99      raw.pileup.df.2$A_IVT = raw.pileup.df$T_IVT
100     raw.pileup.df.2$T_IVT = raw.pileup.df$A_IVT
101     raw.pileup.df.2$C_IVT = raw.pileup.df$G_IVT
102     raw.pileup.df.2$G_IVT = raw.pileup.df$C_IVT
103
```

Create the Processed_pileup folder under init_gene_pileup

Run (source) the scrip

Step4

Specify the data.dir=path to the Processed_pileup folder created in step 3
Specify the k.mer.summary.out.dir=Path to the csv file containing the kmer analysis under the kmer_analysis folder.
Specify the merged.Uonly.Pileup.out.dir cvs file under the finalized_pielup folder.
Change target.cols to D1a, D1b…

Then modify the target.cols according to the replicate names you decided to use.

```
kmer.lst = kmer.lst[order(kmer.lst)]
U.count = rep(0, length(kmer.lst)); names(U.count) = kmer.lst
C.count = rep(0, length(kmer.lst)); names(C.count) = kmer.lst
found.sites = rep(0, length(kmer.lst)); names(found.sites) = kmer.lst

#############################################################################
# path to merged raw pileups for each gene
data.dir                   = "~/CellLinesPVal/SH-SY5Y/data_SH-SY5Y/init_gene_pileup/Processed_pileup/"
#CREATE kmer_analysis AND finalized_pileup FOLDER FIRST!
k.mer.summary.out.dir      = "~/CellLinesPVal/SH-SY5Y/data_SH-SY5Y/init_gene_pileup/kmer_analysis/IVT_kmer_Analysis.csv"
merged.Uonly.Pileup.out.dir = "~/CellLinesPVal/SH-SY5Y/data_SH-SY5Y/init_gene_pileup/finalized_pileup/merged_U_only.csv"

# get the list of the genes in the input directory
genes = list.files(data.dir)

#############################################################################
target.cols = c("Annotation", "chr",        "position",   "strand",
                "target.nucleotide" ,
                "T_rep1",    "C_rep1",
                "T_rep2",    "C_rep2",
                "T_rep3",    "C_rep3",
                "T_IVT",      "C_IVT",    "kmer"   )

```

Create the kmer_analysis and finalized_pileup folders under the init_gene_pileup folder

Run (source) the script

Step5

Provide the path of step4 for kmer.summary and kmer.summary$mm.
Modify the merged.df variable adding the mm filed (mismatches) according to your replicates names.

```
print("Reading kmer summary ... step5")
################################################################################
kmer.summary = read.csv("~/CellLinesPVal/SH-SY5Y/data_SH-SY5Y/init_gene_pileup/kmer_analysis/IVT_kmer_Analysis.csv")
kmer.summary$mm = kmer.summary$C / (kmer.summary$U + kmer.summary$C) * 100

merged.df = read.csv( "~/CellLinesPVal/SH-SY5Y/data_SH-SY5Y/init_gene_pileup/finalized_pileup/merged_U_only.csv")

################################################################################
# calculate mismatches
merged.df$mm.rep1   = merged.df$C_rep1 / (merged.df$T_rep1 + merged.df$C_rep1) * 100
merged.df$mm.rep2   = merged.df$C_rep2 / (merged.df$T_rep2 + merged.df$C_rep2) * 100
merged.df$mm.rep3   = merged.df$C_rep3 / (merged.df$T_rep3 + merged.df$C_rep3) * 100
#merged.df$mm.Diff3   = merged.df$C_Diff3 / (merged.df$T_Diff3 + merged.df$C_Diff3) * 100
# merged.df$mm.Undiff1 = merged.df$C_Undiff1 / (merged.df$T_Undiff1 + merged.df$C_Undiff1) * 100
# merged.df$mm.Undiff2 = merged.df$C_Undiff2 / (merged.df$T_Undiff2 + merged.df$C_Undiff2) * 100
# merged.df$mm.Undiff3 = merged.df$C_Undiff3 / (merged.df$T_Undiff3 + merged.df$C_Undiff3) * 100
# merged.df$mm.WT   = merged.df$C_WT / (merged.df$T_WT + merged.df$C_WT) * 100
merged.df$mm.IVT  = merged.df$C_IVT / (merged.df$T_IVT + merged.df$C_IVT) * 100

################################################################################
# replace 'NA's with 0
merged.df$mm.rep1[which(is.na(merged.df$mm.rep1))] = 0
merged.df$mm.rep2[which(is.na(merged.df$mm.rep2))] = 0
merged.df$mm.rep3[which(is.na(merged.df$mm.rep3))] = 0
#merged.df$mm.Diff3[which(is.na(merged.df$mm.Diff3))] = 0
#merged.df$mm.Undiff1[which(is.na(merged.df$mm.Undiff1))] = 0
#merged.df$mm.Undiff2[which(is.na(merged.df$mm.Undiff2))] = 0
#merged.df$mm.Undiff3[which(is.na(merged.df$mm.Undiff3))] = 0
# merged.df$mm.WT[which(is.na(merged.df$mm.WT))] = 0
merged.df$mm.IVT[which(is.na(merged.df$mm.IVT))] = 0

```

Then add the p.value fields for your replicates (not for IVT)
And add an if statement for each replicate (not for IVT) according to the chosen names.

```
################################################################################
# initialize a column to store calculated p-vlaues
merged.df$p.value.rep1 = 1
merged.df$p.value.rep2 = 1
merged.df$p.value.rep3 = 1
#merged.df$p.value.Diff3 = 1
# merged.df$p.value.Undiff1 = 1
# merged.df$p.value.Undiff2 = 1
# merged.df$p.value.Undiff3 = 1
# merged.df$p.value.WT = 1

min.acc.reads = 8

period = round(nrow(merged.df) / 1000)

for (i in c(1:nrow(merged.df))){#
  # verbose: report progress
  if (i%%period == 0) print(paste0(round(i/nrow(merged.df) * 100, 2),"%"))


################################################################################

  # rep1
  if ((merged.df$C_rep1[i] + merged.df$T_rep1[i]) >= min.acc.reads &  merged.df$mm.rep1[i]>merged.df$expected.mm[i]){
    merged.df$p.value.rep1[i] =
      calc.p.val(n.read = merged.df$C_rep1[i] + merged.df$T_rep1[i],
                 mod.prob = merged.df$expected.mm[i]/100, subject.mm = merged.df$mm.rep1[i]/100 )

  }

  # rep2
  if ((merged.df$C_rep2[i] + merged.df$T_rep2[i]) >= min.acc.reads &  merged.df$mm.rep2[i]>merged.df$expected.mm[i]){
    merged.df$p.value.rep2[i] =
      calc.p.val(n.read = merged.df$C_rep2[i] + merged.df$T_rep2[i],
                 mod.prob = merged.df$expected.mm[i]/100, subject.mm = merged.df$mm.rep2[i]/100 )

  }

  # rep3
  if ((merged.df$C_rep3[i] + merged.df$T_rep3[i]) >= min.acc.reads &  merged.df$mm.rep3[i]>merged.df$expected.mm[i]){
    merged.df$p.value.rep3[i] =
      calc.p.val(n.read = merged.df$C_rep3[i] + merged.df$T_rep3[i],
                 mod.prob = merged.df$expected.mm[i]/100, subject.mm = merged.df$mm.rep3[i]/100 )

  }
```

Change the final line to be shure the path to the finalized_pileup/Merged_with_P_vals.csv file is correct

```
#summary(merged.df[c(1:1000),])
write.csv(merged.df,"~/CellLinesPVal/SH-SY5Y/data_SH-SY5Y/init_gene_pileup/finalized_pileup/Merged_with_P_vals.csv", row.nam
```

Run (source) the script