Reading and Cleaning Data

```
clinical.patients <- read.table("data_clinical_patient.txt", sep = "\t", header = TRUE)
data.mutations <- read.table("data_mutations.txt", sep = "\t", header = TRUE)
data.RNAseq <- read.csv("RNAseq_BRCA.csv")
```

```
library(stringr)

# Rename columns to match the desired format
colnames(data.RNAseq) <- sapply(colnames(data.RNAseq), function(name) {

  segments <- strsplit(name, "\\.")[[1]][1:3]

  paste(segments, collapse = "-")
})
colnames(data.RNAseq)[1] <- "Transcript_ID"
```

```
unique.clinical <- as.data.frame(unique(clinical.patients$PATIENT_ID))
unique.mutations <- as.data.frame(unique(data.mutations$Tumor_Sample_Barcode))
unique.RNA <- as.data.frame(colnames(data.RNAseq[,2:1232]))

mutation.patients <- as.data.frame(data.mutations$Tumor_Sample_Barcode)
colnames(mutation.patients) <- "Patient_ID"
mutation.patients$Patient_ID <- substr(mutation.patients$Patient_ID, 1, 12)

data.mutations <- cbind(mutation.patients, data.mutations)

colnames(data.mutations)
```

```
##   [1] "Patient_ID"                    "Hugo_Symbol"                   "Entrez_Gene_Id"
##   [5] "NCBI_Build"                    "Chromosome"                    "Start_Position"
##   [9] "Strand"                        "Consequence"                   "Variant_Classification"
##  [13] "Reference_Allele"              "Tumor_Seq_Allele1"             "Tumor_Seq_Allele2"
##  [17] "dbSNP_Val_Status"              "Tumor_Sample_Barcode"          "Matched_Norm_Sample_Barcode"
##  [21] "Match_Norm_Seq_Allele2"        "Tumor_Validation_Allele1"      "Tumor_Validation_Allele2"
##  [25] "Match_Norm_Validation_Allele2" "Verification_Status"           "Validation_Status"
##  [29] "Sequencing_Phase"              "Sequence_Source"               "Validation_Method"
##  [33] "BAM_File"                      "Sequencer"                     "t_ref_count"
##  [37] "n_ref_count"                   "n_alt_count"                   "HGVSc"
##  [41] "HGVSp_Short"                   "Transcript_ID"                 "RefSeq"
##  [45] "Codons"                        "Hotspot"                       "AA_MAF"
##  [49] "ALLELE_NUM"                    "AMR_MAF"                       "ASN_MAF"
##  [53] "Amino_acids"                   "BIOTYPE"                       "CANONICAL"
##  [57] "CDS_position"                  "CENTERS"                       "CLIN_SIG"
##  [61] "COSMIC"                        "DBVS"                          "DISTANCE"
##  [65] "EAS_MAF"                       "EA_MAF"                        "ENSP"
##  [69] "EXON"                          "ExAC_AF"                       "ExAC_AF_AFR"
##  [73] "ExAC_AF_EAS"                   "ExAC_AF_FIN"                   "ExAC_AF_NFE"
##  [77] "ExAC_AF_SAS"                   "Existing_variation"            "FILTER"
##  [81] "Feature_type"                  "GENE_PHENO"                    "GMAF"
##  [85] "HGNC_ID"                       "HGVS_OFFSET"                   "HIGH_INF_POS"
##  [89] "INTRON"                        "MERGESOURCE"                   "MOTIF_NAME"
```

```
##  [93] "MOTIF_SCORE_CHANGE"           "NCALLERS"                     "PHENO"
##  [97] "PolyPhen"                     "SAS_MAF"                      "SIFT"
## [101] "SWISSPROT"                    "SYMBOL"                       "SYMBOL_SOURCE"
## [105] "TSL"                          "UNIPARC"                      "VARIANT_CLASS"
## [109] "cDNA_position"                "n_depth"                      "t_depth"
```

```r
colnames(unique.clinical) <- "Patient_ID"
colnames(unique.mutations) <- "Patient_ID"
colnames(unique.RNA) <- "Patient_ID"

unique.mutations$Patient_ID <- substr(unique.mutations$Patient_ID, 1, 12)

# Find common patient IDs across all three data frames
common_patient_ids <- Reduce(intersect, list(unique.clinical$Patient_ID, unique.mutations$Patient_ID, u

filtered.clinical <- clinical.patients[clinical.patients$PATIENT_ID %in% common_patient_ids, ]
filtered.mutations <- data.mutations[data.mutations$Patient_ID %in% common_patient_ids, ]
filtered.RNA <- data.RNAseq[, names(data.RNAseq)%in% common_patient_ids]
filtered.RNA <- cbind(data.RNAseq[,1], filtered.RNA)
colnames(filtered.RNA)[1] <- "Transcript_ID"

#save(filtered.clinical, filtered.mutations, filtered.RNA, file = "cleaning_data.RData")
```