tcga_multiomic

Tarsus Lam

Download multiomic TCGA data

```
library(curatedTCGAData)
```

```
## Loading required package: MultiAssayExperiment
## Loading required package: SummarizedExperiment
## Loading required package: MatrixGenerics
## Loading required package: matrixStats
##
## Attaching package: 'MatrixGenerics'
## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars
## Loading required package: GenomicRanges
## Loading required package: stats4
## Loading required package: BiocGenerics
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
```

```
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:utils':
##
##     findMatches

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

## Loading required package: GenomeInfoDb

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
cohort <- "BRCA"    # Set the TCGA cohort of interest
data_types <- c("RNASeq2GeneNorm", "Mutation", "Methylation_methyl450")    # Specify data types to retr

# Retrieve TCGA data
readData <- curatedTCGAData(cohort, data_types, version = '2.0.1', dry.run = FALSE)
```

```
## Working on: BRCA_Mutation-20160128

## see ?curatedTCGAData and browseVignettes('curatedTCGAData') for documentation

## loading from cache

## require("RaggedExperiment")

## Working on: BRCA_RNASeq2GeneNorm-20160128

## see ?curatedTCGAData and browseVignettes('curatedTCGAData') for documentation

## loading from cache

## Working on: BRCA_Methylation_methyl450-20160128
```

```
## see ?curatedTCGAData and browseVignettes('curatedTCGAData') for documentation
## loading from cache
## require("rhdf5")
## see ?curatedTCGAData and browseVignettes('curatedTCGAData') for documentation
## loading from cache
## Loading required package: HDF5Array
## Loading required package: DelayedArray
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following object is masked from 'package:S4Vectors':
##
##     expand
## Loading required package: S4Arrays
## Loading required package: abind
##
## Attaching package: 'S4Arrays'
## The following object is masked from 'package:abind':
##
##     abind
## The following object is masked from 'package:base':
##
##     rowsum
##
## Attaching package: 'DelayedArray'
## The following objects are masked from 'package:base':
##
##     apply, scale, sweep
##
## Attaching package: 'HDF5Array'
## The following object is masked from 'package:rhdf5':
##
##     h5ls
## Working on: BRCA_colData-20160128
## see ?curatedTCGAData and browseVignettes('curatedTCGAData') for documentation
## loading from cache
## Working on: BRCA_metadata-20160128
## see ?curatedTCGAData and browseVignettes('curatedTCGAData') for documentation
## loading from cache
## Working on: BRCA_sampleMap-20160128
## see ?curatedTCGAData and browseVignettes('curatedTCGAData') for documentation
```

```
## loading from cache
```

```
## harmonizing input:
##   removing 12495 sampleMap rows not in names(experiments)
```

```
readData
```

```
## A MultiAssayExperiment object of 3 listed
##  experiments with user-defined names and respective classes.
##  Containing an ExperimentList class object of length 3:
##  [1] BRCA_Mutation-20160128: RaggedExperiment with 90490 rows and 993 columns
##  [2] BRCA_RNASeq2GeneNorm-20160128: SummarizedExperiment with 20501 rows and 1212 columns
##  [3] BRCA_Methylation_methyl450-20160128: SummarizedExperiment with 485577 rows and 885 columns
## Functionality:
##  experiments() - obtain the ExperimentList instance
##  colData() - the primary/phenotype DataFrame
##  sampleMap() - the sample coordination DataFrame
##  `$`, `[`, `[[` - extract colData columns, subset, or experiment
##  *Format() - convert into a long or wide DataFrame
##  assays() - convert ExperimentList to a SimpleList of matrices
##  exportClass() - save data to flat files
```

```
# Retrieve and examine sample mapping
sample_mapping <- sampleMap(readData)
sample_mapping
```

```
## DataFrame with 3090 rows and 3 columns
##                              assay      primary                colname
##                           <factor>  <character>            <character>
## 1    BRCA_RNASeq2GeneNorm-20160128 TCGA-3C-AAAU TCGA-3C-AAAU-01A-11R..
## 2    BRCA_RNASeq2GeneNorm-20160128 TCGA-3C-AALI TCGA-3C-AALI-01A-11R..
## 3    BRCA_RNASeq2GeneNorm-20160128 TCGA-3C-AALJ TCGA-3C-AALJ-01A-31R..
## 4    BRCA_RNASeq2GeneNorm-20160128 TCGA-3C-AALK TCGA-3C-AALK-01A-11R..
## 5    BRCA_RNASeq2GeneNorm-20160128 TCGA-4H-AAAK TCGA-4H-AAAK-01A-12R..
## ...                           ...          ...                    ...
## 3086        BRCA_Mutation-20160128 TCGA-OL-A66J TCGA-OL-A66J-01A-11D..
## 3087        BRCA_Mutation-20160128 TCGA-OL-A66K TCGA-OL-A66K-01A-11D..
## 3088        BRCA_Mutation-20160128 TCGA-PE-A5DC TCGA-PE-A5DC-01A-12D..
## 3089        BRCA_Mutation-20160128 TCGA-PE-A5DD TCGA-PE-A5DD-01A-12D..
## 3090        BRCA_Mutation-20160128 TCGA-PE-A5DE TCGA-PE-A5DE-01A-11D..
```

```
# Count the number of datasets per sample/patient
dataset_counts <- table(table(sample_mapping$primary))
dataset_counts
```

```
##
##   1   2   3   4   5   6   7   8
##   5 398 580  35  75   2   2   1
```

```
# Examine clinical data
clinical_data <- colData(readData)
head(colnames(clinical_data), 10)
```

```
##  [1] "patientID"             "years_to_birth"        "vital_status"
##  [4] "days_to_death"         "days_to_last_followup" "tumor_tissue_site"
##  [7] "pathologic_stage"      "pathology_T_stage"     "pathology_N_stage"
## [10] "pathology_M_stage"
```

```r
# Analyze pathology_T_stage and create t_stage factor
pathology_t_stage_table <- table(clinical_data$pathology_T_stage)
print(pathology_t_stage_table)
```

```
##
##  t1 t1a t1b t1c  t2 t2a t2b  t3 t3a  t4 t4b t4d  tx
##  41   1  16 223 633   1   1 137   1   9  28   3   3
```

```r
clinical_data$t_stage <- factor(substr(clinical_data$pathology_T_stage, 1, 2))

# Analyze t_stage after removing suffix
t_stage_table <- table(clinical_data$t_stage)
t_stage_table
```

```
##
##  t1  t2  t3  t4  tx
## 281 635 138  40   3
```

```r
# Analyze vital_status table
vital_status_table <- table(clinical_data$vital_status)
vital_status_table
```

```
##
##   0   1
## 945 152
```

```r
# Observe the relationship between t_stage and vital_status
t_stage_vs_vital_status_table <- table(clinical_data$t_stage, clinical_data$vital_status)
t_stage_vs_vital_status_table
```

```
##
##        0   1
##   t1 248  33
##   t2 557  78
##   t3 113  25
##   t4  25  15
##   tx   2   1
```

Process mutation data

```r
# Access mutation data
mutation_data <- readData[[1]]
mutation_data
```

```
## class: RaggedExperiment
## dim: 90490 993
## assays(62): Hugo_Symbol Entrez_Gene_Id ... EVS_AA EVS_All
## rownames: NULL
## colnames(993): TCGA-A1-A0SB-01A-11D-A142-09
##   TCGA-A1-A0SD-01A-11D-A10Y-09 ... TCGA-PE-A5DD-01A-12D-A27P-09
##   TCGA-PE-A5DE-01A-11D-A27P-09
## colData names(0):
```

```r
# Retrieve sample IDs from mutation data
mutation_sample_ids <- colnames(mutation_data)
head(mutation_sample_ids)
```

```
## [1] "TCGA-A1-A0SB-01A-11D-A142-09" "TCGA-A1-A0SD-01A-11D-A10Y-09"
## [3] "TCGA-A1-A0SE-01A-11D-A099-09" "TCGA-A1-A0SF-01A-11D-A142-09"
## [5] "TCGA-A1-A0SG-01A-11D-A142-09" "TCGA-A1-A0SH-01A-11D-A099-09"
```

```
# Display sample IDs from clinical data
head(rownames(clinical_data))
```

```
## [1] "TCGA-A1-A0SB" "TCGA-A1-A0SD" "TCGA-A1-A0SE" "TCGA-A1-A0SF" "TCGA-A1-A0SG"
## [6] "TCGA-A1-A0SH"
```

```
# Truncate to first 12 characters to match clinical sample IDs
mutation_sample_ids <- substr(mutation_sample_ids, 1, 12)

# Check if mutation sample IDs match clinical data
sample_id_match <- all(mutation_sample_ids %in% rownames(clinical_data))
sample_id_match
```

```
## [1] TRUE
```

```
# Display a subset of the mutation data
mutation_subset <- assay(mutation_data)[1:4, 1:4]
mutation_subset
```

```
##                TCGA-A1-A0SB-01A-11D-A142-09 TCGA-A1-A0SD-01A-11D-A10Y-09
## 10:116247760:+ "ABLIM1"                     NA
## 12:43944926:+  "ADAMTS20"                   NA
## 3:85932472:+   "CADM2"                      NA
## 2:25678299:+   "DTNB"                       NA
##                TCGA-A1-A0SE-01A-11D-A099-09 TCGA-A1-A0SF-01A-11D-A142-09
## 10:116247760:+ NA                           NA
## 12:43944926:+  NA                           NA
## 3:85932472:+   NA                           NA
## 2:25678299:+   NA                           NA
```

```
# Count the occurrences of NAs in the mutation data
na_counts <- table(assay(mutation_data)[1,], useNA = "ifany")
na_counts    # almost all NAs
```

```
##
## ABLIM1   <NA>
##      1    992
```

```
# Access mutation assay data per sample instead
mutation_assay <- mutation_data@assays
class(mutation_assay)
```

```
## [1] "CompressedGRangesList"
## attr(,"package")
## [1] "GenomicRanges"
```

```
length(mutation_assay)
```

```
## [1] 993
```

```
mutation_assay_sample <- mutation_assay[[1]]
mutation_symbols <- mutation_assay_sample$Hugo_Symbol
mutation_status <- mutation_assay_sample$Mutation_Status
mutation_classification <- mutation_assay_sample$Variant_Classification
```

```r
# Display tables for mutation information
table(mutation_symbols)
```

```
## mutation_symbols
##          ABLIM1        ADAMTS20           CADM2            DTNB ENSG00000267261
##               1               1               1               1               1
##            MSH3             MYB         NPIPL2          OR11H1            OTOR
##               1               1               1               1               1
##          P2RY10          PIEZO1          SLC6A9          SOX15            SPTB
##               1               1               1               1               1
##         TMEM247          ZNF566          ZNF574          ZNF777
##               1               1               1               1
```

```r
table(mutation_status)
```

```
## mutation_status
## Somatic
##      19
```

```r
table(mutation_classification)
```

```
## mutation_classification
##   Frame_Shift_Del   Frame_Shift_Ins Missense_Mutation            Silent
##                 3                 1                13                 2
```

```r
# Create a single dataframe for mutation data
mut_df = mapply(function(id, a) {
    d = as.data.frame(mcols(a)[c("Hugo_Symbol", "Variant_Classification")])
    names(d) = c("symbol", "variant_class")
    d$patientID = id
    d
}, id = mutation_sample_ids, a = mutation_assay, SIMPLIFY = FALSE, USE.NAMES = FALSE)
mutation_df = do.call(rbind, mut_df)
head(mutation_df)
```

```
##            symbol     variant_class    patientID
## 1          ABLIM1 Missense_Mutation TCGA-A1-A0SB
## 2        ADAMTS20 Missense_Mutation TCGA-A1-A0SB
## 3           CADM2            Silent TCGA-A1-A0SB
## 4            DTNB Missense_Mutation TCGA-A1-A0SB
## 5 ENSG00000267261 Missense_Mutation TCGA-A1-A0SB
## 6            MSH3   Frame_Shift_Del TCGA-A1-A0SB
```

```r
# Create a table for mutation symbols and variant classifications
mutation_table <- table(mutation_df$symbol, mutation_df$variant_class)

# Calculate the total number of specific mutation types
mutation_types <- c("Missense_Mutation", "Nonsense_Mutation", "Frame_Shift_Del", "Frame_Shift_Ins")
mutation_totals <- apply(mutation_table[, mutation_types], 1, sum)

# Order mutation symbols by the total number of mutations
mutation_order <- order(mutation_totals, decreasing = TRUE)
top_mutations <- mutation_table[mutation_order[1:10], mutation_types]
top_mutations
```

```
##
##          Missense_Mutation Nonsense_Mutation Frame_Shift_Del Frame_Shift_Ins
```

```
##    PIK3CA              374                0               0               2
##    TP53                198               47              43              13
##    TTN                 257               11              12               5
##    MUC16               103                5               1               0
##    CDH1                 20               31              27              30
##    MAP3K1               18               14              33              33
##    GATA3                10                2              18              53
##    MLL3                 29               23              18              13
##    MUC12                68                3               3               9
##    MUC4                 59                1               1               1
```

Combine mutation and clinical data

```
# Calculate the number of mutations per patient
nmut <- sapply(split(mutation_df$patientID, mutation_df$patientID), length)

# Display the first few values
head(nmut)
```

```
## TCGA-A1-A0SB TCGA-A1-A0SD TCGA-A1-A0SE TCGA-A1-A0SF TCGA-A1-A0SG TCGA-A1-A0SH
##           19           27           27           40           34           94
```

```
# Determine overlapping information
nmut_length <- length(nmut)
clin_rows <- nrow(clinical_data)
overlap_check <- all(names(nmut) %in% rownames(clinical_data))
clin_mut <- clinical_data[names(nmut),]

# Display the results
nmut_length
```

```
## [1] 977
```

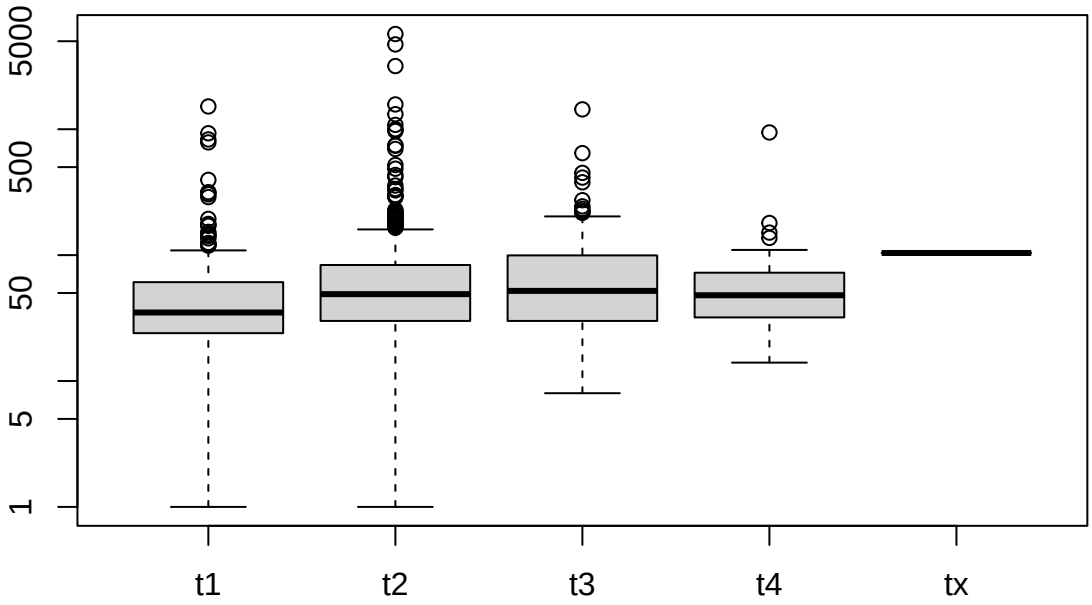
```
clin_rows
```

```
## [1] 1098
```

```
overlap_check
```

```
## [1] TRUE
```

```
# Create a boxplot of mutations per tumor stage
with(clin_mut, boxplot(split(nmut, t_stage), log = "y"))
```




```
# Combine patient information and TP53 mutation presence
tp53_mut_pts <- mutation_df[mutation_df$symbol == "TP53", "patientID"]
clin_mut$tp53_mut <- clin_mut$patientID %in% tp53_mut_pts

# Create a table to show TP53 mutation presence by tumor stage
table(clin_mut$tp53_mut, clin_mut$t_stage)    # TP53 most common in t2
```

```
##
##          t1  t2  t3  t4  tx
##   FALSE 187 372  91  25   0
##   TRUE   70 192  29  10   1
```

Combine expression and clinical data

```
library(limma)
```

```
##
## Attaching package: 'limma'
```

```
## The following object is masked from 'package:BiocGenerics':
##
##     plotMA
```

```
# Access RNA-Seq data
rnaseq <- readData[[2]]
rnaseq
```

```
## class: SummarizedExperiment
## dim: 20501 1212
## metadata(3): filename build platform
## assays(1): ''
## rownames(20501): A1BG A1CF ... psiTPTE22 tAKR
## rowData names(0):
## colnames(1212): TCGA-3C-AAAU-01A-11R-A41B-07
##   TCGA-3C-AALI-01A-11R-A41B-07 ... TCGA-Z7-A8R5-01A-42R-A41B-07
##   TCGA-Z7-A8R6-01A-11R-A41B-07
```

```
## colData names(0):
```

```
assay(rnaseq)[1:3, 1:3]
```

```
##       TCGA-3C-AAAU-01A-11R-A41B-07 TCGA-3C-AALI-01A-11R-A41B-07
## A1BG                      197.0897                     237.3844
## A1CF                        0.0000                       0.0000
## A2BP1                       0.0000                       0.0000
##       TCGA-3C-AALJ-01A-31R-A41B-07
## A1BG                      423.2366
## A1CF                        0.9066
## A2BP1                       0.0000
```

```
# Perform log2(x+1) transformation
assay(rnaseq) <- log2(assay(rnaseq) + 1)
assay(rnaseq)[1:3, 1:3]
```

```
##       TCGA-3C-AAAU-01A-11R-A41B-07 TCGA-3C-AALI-01A-11R-A41B-07
## A1BG                       7.63001                     7.897146
## A1CF                       0.00000                     0.000000
## A2BP1                      0.00000                     0.000000
##       TCGA-3C-AALJ-01A-31R-A41B-07
## A1BG                     8.7287253
## A1CF                     0.9310022
## A2BP1                    0.0000000
```

```
# Shorten column names to match clinical data
colnames(rnaseq) <- substr(colnames(rnaseq), 1, 12)

# Append clinical data to RNA-Seq data
colData(rnaseq) <- clinical_data[colnames(rnaseq),]

# Treat 't_stage' as numeric and perform differential expression analysis
rnaseq$numts <- as.numeric(factor(rnaseq$t_stage))
mm <- model.matrix(~numts, data=colData(rnaseq))
f1 <- lmFit(assay(rnaseq), mm)
ef1 <- eBayes(f1)
```

```
## Warning: Zero sample variances detected, have been offset away from zero
```

```
top_genes <- topTable(ef1, n=20)[topTable(ef1, n=20)$adj.P.Val <= 0.05,]
```

```
## Removing intercept from test coefficients
```

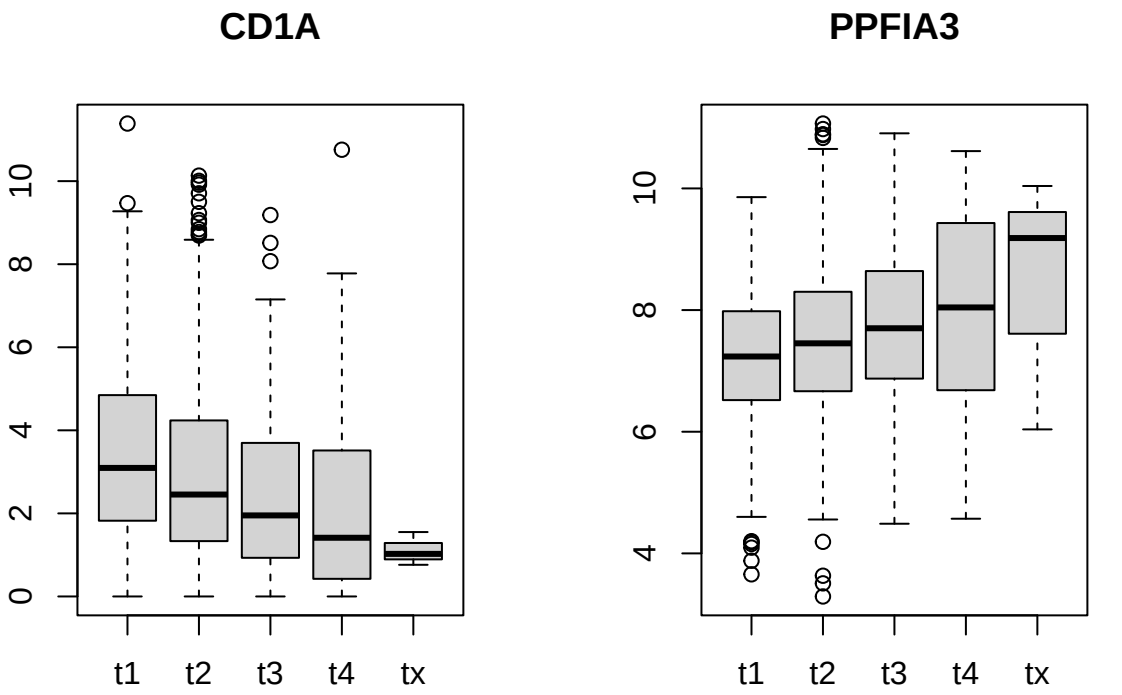
```
## Removing intercept from test coefficients
```

```
# Display the top differentially expressed genes
top_genes
```

```
##                logFC     AveExpr         t      P.Value   adj.P.Val        B
## CD1B      -0.36357996  1.96517981 -5.464469 5.631830e-08 0.001109057 7.961704
## CD207     -0.41033800  4.44592272 -5.321568 1.224769e-07 0.001109057 7.230500
## IKZF4     -0.11945341  7.74749239 -5.268940 1.622931e-07 0.001109057 6.965860
## CD1A      -0.43745683  2.96472127 -5.199836 2.339732e-07 0.001199171 6.622183
## C12orf35  -0.15699534 10.22033258 -5.139871 3.202603e-07 0.001313131 6.327470
## ERMN      -0.32583912  4.25124840 -5.059918 4.842685e-07 0.001654665 5.939608
## CD1E      -0.36720871  4.27154436 -5.008909 6.285559e-07 0.001699300 5.695199
## COG4       0.09605496 10.04515183  4.990360 6.906765e-07 0.001699300 5.606911
```

```
## ZBTB1     -0.09159556  9.70491159 -4.958735 8.104901e-07 0.001699300 5.457108
## TLR10     -0.31009480  4.49029725 -4.954284 8.288866e-07 0.001699300 5.436095
## ZMYM6     -0.07544481  8.81003125 -4.906031 1.055974e-06 0.001966074 5.209481
## C20orf166  0.08459620  0.04327255  4.886591 1.163478e-06 0.001966074 5.118788
## ARAF       0.08441422 10.25521623  4.869958 1.263763e-06 0.001966074 5.041464
## CCR6      -0.26015940  5.81212617 -4.857749 1.342619e-06 0.001966074 4.984868
## CD1C      -0.35969558  4.47587154 -4.802583 1.761982e-06 0.002316110 4.730836
## MIAT      -0.25209288  7.00316907 -4.785488 1.915751e-06 0.002316110 4.652682
## LOC646999 -0.17590812  3.77058804 -4.784972 1.920583e-06 0.002316110 4.650329
## ZNF267    -0.10448768  8.52868803 -4.726182 2.555662e-06 0.002874175 4.383634
## FCRL4     -0.21585596  1.15738878 -4.711584 2.742211e-06 0.002874175 4.317901
## PPFIA3     0.22873933  7.49475257  4.706963 2.803937e-06 0.002874175 4.297135
```

```
# Examples of associated genes
par(mfrow = c(1, 2))
boxplot(split(assay(rnaseq)["CD1A", ], rnaseq$t_stage), main = "CD1A")    # Higher expression in lower
boxplot(split(assay(rnaseq)["PPFIA3", ], rnaseq$t_stage), main = "PPFIA3")    # Higher expression in hi
```



Combine methylation and expression data

```
library(curatedTCGAData)

# Access the methylation data
methyl <- readData[[3]]
methyl
```

```
## class: SummarizedExperiment
## dim: 485577 885
## metadata(0):
## assays(1): counts
## rownames(485577): cg00000029 cg00000108 ... rs966367 rs9839873
## rowData names(3): Gene_Symbol Chromosome Genomic_Coordinate
```

```
## colnames(885): TCGA-3C-AAAU-01A-11D-A41Q-05
##   TCGA-3C-AALI-01A-11D-A41Q-05 ... TCGA-Z7-A8R5-01A-42D-A41Q-05
##   TCGA-Z7-A8R6-01A-11D-A41Q-05
## colData names(0):
```

```
assay(methyl)
```

```
## <485577 x 885> DelayedMatrix object of type "double":
##            TCGA-3C-AAAU-01A-11D-A41Q-05 ... TCGA-Z7-A8R6-01A-11D-A41Q-05
## cg00000029                   0.10362281   .                   0.07741195
## cg00000108                           NA   .                           NA
## cg00000109                           NA   .                           NA
## cg00000165                   0.09736179   .                   0.07340964
## cg00000236                   0.87820501   .                   0.89658236
##        ...                            .   .                            .
##  rs9363764                    0.2104274   .                   0.54695352
##   rs939290                    0.5788985   .                   0.02594884
##   rs951295                    0.9459935   .                   0.54311380
##   rs966367                    0.4181337   .                   0.50595456
##  rs9839873                    0.7395188   .                   0.94395293
```

```
# Filter for primary tumor tissue samples
isprimary <- sapply(strsplit(colnames(methyl), split = "-"), '[[', 4) == "01A"
methyl <- methyl[, isprimary]

# Shorten column names to match clinical data
colnames(methyl) <- substr(colnames(methyl), 1, 12)

# Append clinical data to methylation data
colData(methyl) <- clinical_data[colnames(methyl),]

# Check for sufficient samples for analysis
intersect_samples <- length(intersect(colnames(methyl), colnames(rnaseq)))

# Subset the intersection between Methylation and RNA-Seq samples
methyl_subset <- methyl[, which(colnames(methyl) %in% colnames(rnaseq))]
rnaseq_subset <- rnaseq[, which(colnames(rnaseq) %in% colnames(methyl))]

# Replace duplicate columns with row means
duplicates <- unique(colnames(rnaseq_subset)[duplicated(colnames(rnaseq_subset))])
mean_vals <- sapply(duplicates, function(col) {
  rowMeans(assay(rnaseq_subset)[, colnames(rnaseq_subset) == col])
})
rnaseq_subset <- rnaseq_subset[, !duplicated(colnames(rnaseq_subset))]

# Check for sample and order consistency
identical_samples <- identical(row.names(assay(rnaseq_subset)), row.names(mean_vals))
assay(rnaseq_subset)[, duplicates] <- mean_vals
identical_order <- identical(colnames(rnaseq_subset), colnames(methyl_subset))

# Extract methylation genes
methyl_genes <- rowData(methyl_subset)$Gene_Symbol
methyl_genes <- methyl_genes[!is.na(methyl_genes)]

# Display the first few methylation genes
```

```r
head(methyl_genes)
```

```
## [1] "RBL2"    "C3orf35" "FNDC3B"  "VDAC3"   "ACTN1"   "ATP2A1"
```

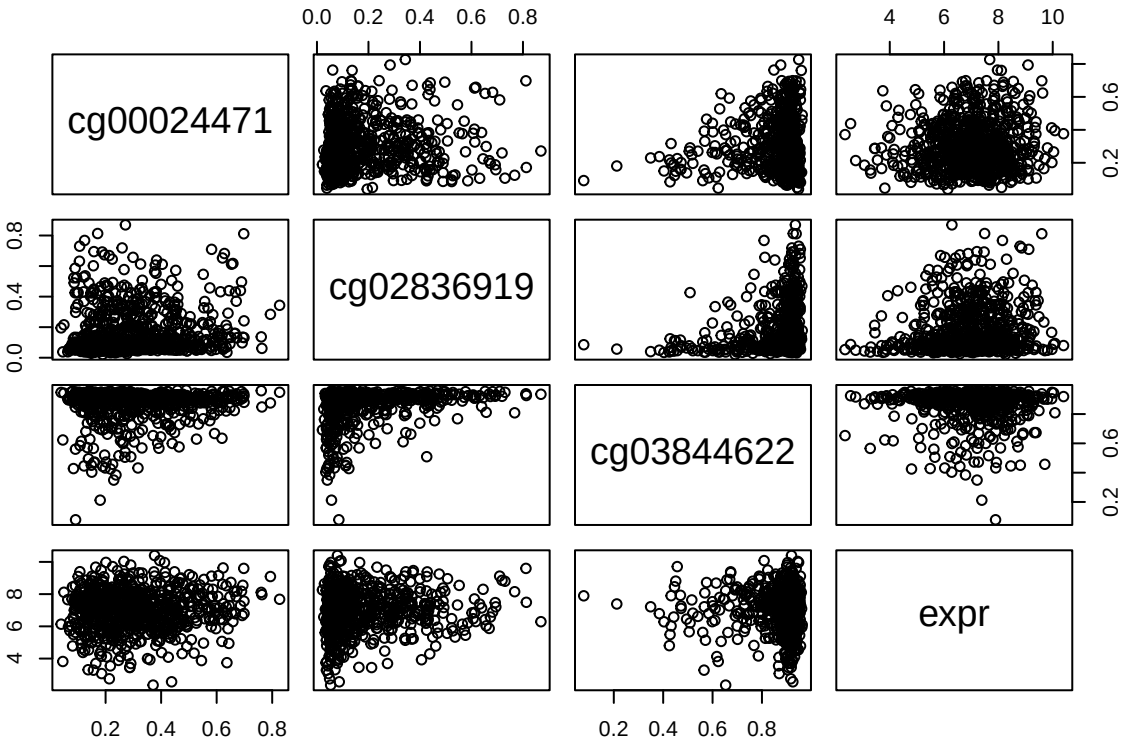
```r
# Function to calculate correlation between methylation and expression data
meth_rna_corr <- function(sym, mpick = 3) {
  # Subset to the first mpick methylation sites for the given gene symbol
  methyl_ind <- which(methyl_genes == sym)
  if (length(methyl_ind) > mpick) {
    methyl_ind <- methyl_ind[1:mpick]
  }
  methyl_dat <- assay(methyl_subset)[methyl_ind,]

  # Subset expression data to the selected gene symbol
  expr_ind <- which(rownames(rnaseq_subset) == sym)
  expr_dat <- assay(rnaseq_subset)[expr_ind,]

  # Combine methylation and expression data as a data frame
  combined_dat <- data.frame(t(methyl_dat), expr = expr_dat)

  # Plot pairs and calculate correlation coefficients between methylation and expression
  pairs(combined_dat)
  correlations <- sapply(1:mpick, function(i) {
    cor(as.numeric(combined_dat[, i]), combined_dat$expr)
  })
  correlations
}

# Calculate correlation for given gene with specified number of methylation sites
gene_of_interest <- 'BRCA2'
num_sites <- 3
meth_rna_corr(gene_of_interest, num_sites)
```




```
## [1]  0.07345558  0.09608981 -0.04273168
```