Introduction

Introduction

These instructions are aimed at people familiar with R and familiar with TCGA/GDC platforms and data types. They are intended to introduce the reader to producing the given assessment. These instructions will only rarely, if ever, touch on the appropriateness of the assessment algorithm or interpretation of output. See MBatch_01_InstallLinux for instructions on downloading test data.

Algorithm

EBNPlus_CheckData_Structures is a supplementary function for use with the EBNPlus correction. This function takes two matrices and checks that they will work as arguments to other MBatch EBNPlus functions. This function checks the following: *Both matrix arguments pass is.matrix test. *Both matrices have column names. *Both matrices have row names and they interect at least once. *Both matrices have column names that intersect at least once or the replicate value vectors are the same size and exist in the column names. *All data in the matrices is numeric.

Output

This function produces no particular output, but performs a stop if the check conditions are not met.

Usage

EBNPlus_CheckData_Structures(theDataMatrix1, theDataMatrix2, theDataReplicates1 = NULL, theDataReplicates2 = NULL)

Arguments

##theDataMatrix1
A matrix for data set 1 containing numeric values with columns being sample ids and rows being gene ids.

##theDataMatrix2
A matrix for data set 2 containing numeric values with columns being sample ids and rows being gene ids.

##theDataReplicates1
A vector of "replicates" in data set 1 used for corrections. Defaults to NULL.

##theDataReplicates2
A vector of "replicates" in data set 2 used for corrections. Defaults to NULL.

Example Call

The following code is adapted from the tests/EBNPlus_Correction_Structures.R file. Data used is from the testing data as per the MBatch_01_InstallLinux document.

```
{
  library(MBatch)

  inputDir <- getTestInputDir()
  outputDir <- getTestOutputDir()
  compareDir <- getTestCompareDir()

  # set the paths
  theDataFile1=file.path(inputDir, "brca_rnaseq2_matrix_data.tsv")
  theDataFile2=file.path(inputDir, "brca_agi4502_matrix_data.tsv")

  # trim genes to get just gene symbols from standardized data
  trimGenes <- function(theGenes)
  {
    foo <- as.vector(unlist(
      sapply(theGenes, function(theGene)
      {
        # keep the same if it starts with ?
        if (TRUE==grepl("^[?]+", theGene))
        {
          return(theGene)
        }
        else
        {
          # split on the | and take the first argument
```

```
        # this makes no change if no pipe
        return(strsplit(theGene, "|", fixed=TRUE)[[1]][1])
      }
    })
  ))
  foo
}

# remove duplicates from columns (samples)
removeDuplicatesFromColumns <- function(theMatrix)
{
  indexOfDuplicates <- which(duplicated(colnames(theMatrix)))
  if (length(indexOfDuplicates) > 0)
  {
    # minus sign uses inverse of indexes
    theMatrix <- theMatrix[ ,-indexOfDuplicates]
  }
  return(theMatrix)
}

# remove duplicates from rows (genes/probes)
removeDuplicatesFromRows <- function(theMatrix)
{
  indexOfDuplicates <- which(duplicated(rownames(theMatrix)))
  if (length(indexOfDuplicates) > 0)
  {
    # minus sign uses inverse of indexes
    theMatrix <- theMatrix[-indexOfDuplicates, ]
  }
  return(theMatrix)
}

if ((!dir.exists(theDataFile1))&&(!dir.exists(theDataFile2)))
{
  warnLevel<-getOption("warn")
  on.exit(options(warn=warnLevel))
  # warnings are errors
  options(warn=3)
  # if there is a warning, show the calls leading up to it
  options(showWarnCalls=TRUE)
  # if there is an error, show the calls leading up to it
  options(showErrorCalls=TRUE)
  #
  # read the files in. This can be done however you want
  theDataMatrix1 <- readAsGenericMatrix(theDataFile1)
  theDataMatrix2 <- readAsGenericMatrix(theDataFile2)
```

```
    # this is the reduce genes to just gene symbols, handling those from standardized data
    rownames(theDataMatrix1) <- trimGenes(rownames(theDataMatrix1))
    rownames(theDataMatrix2) <- trimGenes(rownames(theDataMatrix2))
    # remove any duplicates (this is a requirement for EBNplus)
    theDataMatrix1 <- removeDuplicatesFromColumns(removeDuplicatesFromRows(theDataMatrix1))
    theDataMatrix2 <- removeDuplicatesFromColumns(removeDuplicatesFromRows(theDataMatrix2))
    print("Is this data acceptable?")
    EBNPlus_CheckData_Structures(theDataMatrix1, theDataMatrix2)
    print("If you see this, it is.")
  }
}
```

```
## [1] "Is this data acceptable?"
## [1] "If you see this, it is."
```