

Using MBatch Corrections: EBNPlus__TrainAndValidateReplicates__Structures

Tod Casasent

2023-10-06

1 Use EBNPlus__Correction__Structures for Corrections

Use the EBNPlus__Correction__Structures function to performed corrections.

For most users, the function EBNPlus__Correction__Structures is what you want to use for processing. It is designed specifically to do corrections. The EBNPlus__TrainAndValidate functions are for researchers interested in the internal workings of the EBNPlus algorithm.

See the tests/EBNPlus__Correction__Structures.R or tests/EBNPlus__Correction__Files.R for details.

2 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.

3 Algorithm

EBNPlus__TrainAndValidateReplicates__Structures creates and returns a training and validation set and corrected data, based on the training and validation set.

4 Output

The primary output method for MBatch is to view results in the Batch Effects Website. Correction algorithms generally do not create graphical output and instead create TSV output files.

5 Usage

```
EBNPlus__TrainAndValidateReplicates__Structures(theDataMatrix1, theDataMatrix2, theBatchId1, the-  
BatchId2, theEBNP__BatchWithZero, theEBNP__FixDataSet, theEBNP__CorrectForZero, theEBNP__ParametricPriorsFlag,  
theEBNP__ValidationRatio, theEBNP__TestRatio=0, theSeed = NULL, theEBNP__PriorPlotsFile=NULL,  
theEBNP__MinSampleNum = 3, theEBNP__AddData1Rows = FALSE, theEBNP__AddData2Rows = FALSE,  
theTestSeed=NULL)
```

6 Arguments

##theDataMatrix1

A matrix for data set 1 containing numeric values with columns being sample ids and rows being feature (like

gene or probe) ids.

##theDataMatrix2

A matrix for data set 2 containing numeric values with columns being sample ids and rows being feature (like gene or probe) ids.

##theBatchId1

The group name for data set 1 (for example, RNASeqV2)

##theBatchId2

The group name for data set 2 (for example, Agilent4502)

##theEBNP_BatchWithZero Batch With Zero indicates which data set contains zero values. This is a string “1”, “2”, or “both”. These values will be changed in a future release.

##theEBNP_FixDataSet

Fix Data Set indicates which data set should be set as invariate during the corrections. Value are numeric, 1 or 2 for the appropriate data set, or as.numeric(NA) for neither.

##theEBNP_CorrectForZero

Correct For Zero indicates whether or not data sets marked as “Batch With Zero” should be modified from zero (TRUE or FALSE)

##theEBNP_ParametricPriorsFlag

Use parametric adjustments for corrections (TRUE or FALSE)

##theEBNP_ValidationRatio

A numeric. If 0, do not use a training set. For this function, the value should be less than 1. 0.33 means 33 percent of the replicates available are used for validation with 67 percent for training.

##theEBNP_TestRatio A numeric. If 0, do not use a test set. For this function, the value should be less than 1. 0.33 means 33 percent of the replicates available are used for the test set with 67 percent for training.

##theSeed

NULL or a seed to use. Defaults to NULL. This is passed to the standard R set.seed function.

##theEBNP_PriorPlotsFile

Defaults to NULL and does not create PNG. Non-NULL should be the full path and filename for where to create the prior plots PNG.

##theEBNP_MinSampleNum

Defaults to 3. Any row (gene) with less than this number of samples is dropped.

##theEBNP_AddData1Rows

Defaults to FALSE. TRUE indicates rows in data set 2 not in 1 should be added as all NA.

##theEBNP_AddData2Rows

Defaults to FALSE. TRUE indicates rows in data set 1 not in 2 should be added as all NA.

##theTestSeed

NULL or a seed to use. Defaults to NULL. This is passed to the standard R set.seed function. This is used only when creating Test Set.

7 Example Call

The following code is adapted from the tests/EB_withNonParametricPriors.R file. Data used is from the testing data as per the MBatch_01_InstallLinux document. In the future, we plan to make the output from MBatch more user friendly, but currently, this produces the following output at the command line.

```
{  
  require(MBatch)
```

```

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

theDataFile1=cleanFilePath(inputDir, "brca_rnaseq2_matrix_data.tsv")
theDataFile2=cleanFilePath(inputDir, "brca_agi4502_matrix_data.tsv")
theOutputDir=cleanFilePath(outputDir, "EBNPlus_TrainAndValidateReplicates_Structures1")
theBatchId1="RNASeqV2"
theBatchId2="Agilent4502"
theRandomSeed=314

# 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)
{
  indexOfDuplications <- which(duplicated(colnames(theMatrix)))
  if (length(indexOfDuplications) > 0)
  {
    # minus sign uses inverse of indexes
    theMatrix <- theMatrix[ ,-indexOfDuplications]
  }
  return(theMatrix)
}

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

```

```

    }
    return(theMatrix)
}

printMatrix <- function(theMatrix)
{
  print(is.matrix(theMatrix))
  print(dim(theMatrix))
  rowMax <- dim(theMatrix)[1]
  colMax <- dim(theMatrix)[2]
  rowMax <- min(rowMax, 4)
  colMax <- min(colMax, 4)
  print(theMatrix[1:rowMax, 1:colMax])
}

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)
  #
  unlink(theOutputDir, recursive=TRUE)
  dir.create(theOutputDir, showWarnings=FALSE, recursive=TRUE)
  # read the files in. This can be done however you want
  print("read the files")
  theDataMatrix1 <- readAsGenericMatrix(theDataFile1)
  theDataMatrix2 <- readAsGenericMatrix(theDataFile2)
  # this is the reduce genes to just gene symbols, handling those from standardized data
  print("reduce to gene symbols")
  rownames(theDataMatrix1) <- trimGenes(rownames(theDataMatrix1))
  rownames(theDataMatrix2) <- trimGenes(rownames(theDataMatrix2))
  # remove any duplicates (this is a requirement for EBNplus)
  print("remove duplicates")
  theDataMatrix1 <- removeDuplicatesFromColumns(removeDuplicatesFromRows(theDataMatrix1))
  theDataMatrix2 <- removeDuplicatesFromColumns(removeDuplicatesFromRows(theDataMatrix2))
  print("EBNplus_TrainAndValidateReplicates_Structures")
  resultsList <- EBNplus_TrainAndValidateReplicates_Structures(
    theDataMatrix1, theDataMatrix2, theBatchId1, theBatchId2,
    theEBNP_BatchWithZero="1",
    theEBNP_FixDataSet=as.numeric(NA),
    theEBNP_CorrectForZero=TRUE,
    theEBNP_ParametricPriorsFlag=TRUE,
    theEBNP_ValidationRatio=0.3,
    theEBNP_TestRatio=0.3,
    theSeed=theRandomSeed,
    theTestSeed=theRandomSeed,
    thePriorPlotPath=theOutputDir,

```

```

    theDataVersion="DATA_2022-09-09-1600",
    theTestVersion="TEST_2022-10-10-1300",
    thePriorPlotFile="priorplots.PNG")
print("TestSet1")
printMatrix(resultsList$TestSet1)
print("TestSet2")
printMatrix(resultsList$TestSet2)
print("TrainingSet1")
printMatrix(resultsList$TrainingSet1)
print("TrainingSet2")
printMatrix(resultsList$TrainingSet2)
print("TrainingResults")
printMatrix(resultsList$TrainingResults)
print("ValidationSet1")
printMatrix(resultsList$ValidationSet1)
print("ValidationSet2")
printMatrix(resultsList$ValidationSet2)
print("ValidationResults")
printMatrix(resultsList$ValidationResults)
print("CorrectedResults")
printMatrix(resultsList$CorrectedResults)
}
}

```

```

## [1] "read the files"
## [1] "reduce to gene symbols"
## [1] "remove duplicates"
## [1] "EBNPlus_TrainAndValidateReplicates_Structures"
## 2023 10 06 12:34:08.995 DEBUG qcprludev10 starting BeaEBNPlus
## 2023 10 06 12:34:08.995 DEBUG qcprludev10 MBatch Version: BEA_VERSION_TIMESTAMP
## 2023 10 06 12:34:08.996 DEBUG qcprludev10 BeaEBNPlus theEBNP_AddData1Rows= FALSE
## 2023 10 06 12:34:08.996 DEBUG qcprludev10 BeaEBNPlus theEBNP_AddData2Rows= FALSE
## 2023 10 06 12:34:08.997 DEBUG qcprludev10 priorPlotFile= /BEA/BatchEffectsPackage_data/testing_dynam
## 2023 10 06 12:34:08.998 DEBUG qcprludev10 EBNPlus theEBNP_AddData1Rows= FALSE
## 2023 10 06 12:34:08.998 DEBUG qcprludev10 EBNPlus theEBNP_AddData2Rows= FALSE
## 2023 10 06 12:34:08.999 DEBUG qcprludev10 EBNPlus theEBNP_PriorPlotsFile= /BEA/BatchEffectsPackage_d
## 2023 10 06 12:34:08.999 DEBUG qcprludev10 dim(theData1)=10000 1215
## 2023 10 06 12:34:08.999 DEBUG qcprludev10 dim(theData2)=10000 600
## 2023 10 06 12:34:09.000 DEBUG qcprludev10 remove unknown genes, that start with ?
## 2023 10 06 12:34:09.104 DEBUG qcprludev10 check on adding missing rows
## 2023 10 06 12:34:09.105 DEBUG qcprludev10 dim(theData1)=10000 1215
## 2023 10 06 12:34:09.106 DEBUG qcprludev10 dim(theData2)=10000 600
## 2023 10 06 12:34:09.106 DEBUG qcprludev10 make EBNplus
## 2023 10 06 12:34:09.107 DEBUG qcprludev10 before callNextMethod
## 2023 10 06 12:34:09.110 DEBUG qcprludev10 dim(mData1)=10000 1215
## 2023 10 06 12:34:09.111 DEBUG qcprludev10 dim(mData2)=10000 600
## 2023 10 06 12:34:09.111 DEBUG qcprludev10 removing column duplicates
## 2023 10 06 12:34:09.112 DEBUG qcprludev10 removeDuplicatesFromColumns
## 2023 10 06 12:34:09.112 DEBUG qcprludev10 removeDuplicatesFromColumns
## 2023 10 06 12:34:09.113 DEBUG qcprludev10 dim(mData1)=10000 1215
## 2023 10 06 12:34:09.113 DEBUG qcprludev10 dim(mData2)=10000 600
## 2023 10 06 12:34:09.114 DEBUG qcprludev10 removing row duplicates
## 2023 10 06 12:34:09.114 DEBUG qcprludev10 removeDuplicatesFromRows
## 2023 10 06 12:34:09.115 DEBUG qcprludev10 removeDuplicatesFromRows

```

```

## 2023 10 06 12:34:09.115 DEBUG qcprludev10 dim(mData1)=10000 1215
## 2023 10 06 12:34:09.116 DEBUG qcprludev10 dim(mData2)=10000 600
## 2023 10 06 12:34:09.116 DEBUG qcprludev10 after callNextMethod
## 2023 10 06 12:34:09.116 DEBUG qcprludev10 dim(mData1)=10000 1215
## 2023 10 06 12:34:09.117 DEBUG qcprludev10 dim(mData2)=10000 600
## 2023 10 06 12:34:09.117 DEBUG qcprludev10 after EBNplus
## 2023 10 06 12:34:09.117 DEBUG qcprludev10 dim(ebObj@mData1)=10000 1215
## 2023 10 06 12:34:09.118 DEBUG qcprludev10 dim(ebObj@mData2)=10000 600
## 2023 10 06 12:34:09.120 DEBUG qcprludev10 getBiComOrder
## 2023 10 06 12:34:09.120 DEBUG qcprludev10 makeCommonRows
## 2023 10 06 12:34:09.206 DEBUG qcprludev10 dim(mData1)=7785 1215
## 2023 10 06 12:34:09.206 DEBUG qcprludev10 dim(mData2)=7785 600
## 2023 10 06 12:34:09.207 DEBUG qcprludev10 makeCommonCols
## 2023 10 06 12:34:09.254 DEBUG qcprludev10 dim(mData1)=7785 586
## 2023 10 06 12:34:09.255 DEBUG qcprludev10 dim(mData2)=7785 586
## 2023 10 06 12:34:09.255 DEBUG qcprludev10 asSameOrder
## 2023 10 06 12:34:09.255 DEBUG qcprludev10 asSameOrder before
## 2023 10 06 12:34:09.256 DEBUG qcprludev10 asSameOrder after if 1
## 2023 10 06 12:34:09.273 DEBUG qcprludev10 asSameOrder after if 2
## 2023 10 06 12:34:09.274 DEBUG qcprludev10 row.names(mat1)
## 2023 10 06 12:34:09.275 DEBUG qcprludev10 rownames(mat1)
## 2023 10 06 12:34:09.275 DEBUG qcprludev10 row.names(mat2)
## 2023 10 06 12:34:09.275 DEBUG qcprludev10 rownames(mat2)
## 2023 10 06 12:34:09.276 DEBUG qcprludev10 m.i
## 2023 10 06 12:34:09.458 DEBUG qcprludev10 asSameOrder after if 3
## 2023 10 06 12:34:09.459 DEBUG qcprludev10 dim(ebObj@mat1Com)=7785 586
## 2023 10 06 12:34:09.459 DEBUG qcprludev10 dim(ebObj@mat2Com)=7785 586
## 2023 10 06 12:34:09.460 DEBUG qcprludev10 getValidationSet
## 2023 10 06 12:34:09.460 INFO qcprludev10 getValidationSet seed= 314
## 2023 10 06 12:34:09.461 DEBUG qcprludev10 validationRatio=0.3
## 2023 10 06 12:34:09.461 DEBUG qcprludev10 numSamples=586
## 2023 10 06 12:34:09.485 DEBUG qcprludev10 dim(ebObj@mat1Validation)=7785 175
## 2023 10 06 12:34:09.486 DEBUG qcprludev10 dim(ebObj@mat2Validation)=7785 175
## 2023 10 06 12:34:09.486 DEBUG qcprludev10 dim(ebObj@mat1Train)=7785 411
## 2023 10 06 12:34:09.487 DEBUG qcprludev10 dim(ebObj@mat2Train)=7785 411
## 2023 10 06 12:34:09.487 DEBUG qcprludev10 before Test Set
## 2023 10 06 12:34:09.511 DEBUG qcprludev10 after Test Set
## 2023 10 06 12:34:09.555 DEBUG qcprludev10 train
## 2023 10 06 12:34:09.555 DEBUG qcprludev10 inside train
## 2023 10 06 12:34:09.556 DEBUG qcprludev10 train theEBNP_PriorPlotsFile= /BEA/BatchEffectsPackage_data
## 2023 10 06 12:34:09.570 DEBUG qcprludev10 colnames(Object@mat1Train)=TCGA-A1-AOSH-01A-11R-A084-07.RN
## 2023 10 06 12:34:09.570 DEBUG qcprludev10 colnames(Object@mat2Train)=TCGA-A1-AOSH-01A-11R-A084-07.Ag
## 2023 10 06 12:34:09.570 DEBUG qcprludev10 inside train, call getData4EB
## 2023 10 06 12:34:09.906 DEBUG qcprludev10 data4EB <- as.matrix
## 2023 10 06 12:34:09.907 DEBUG qcprludev10 dim(data4EB)=7785 576
## 2023 10 06 12:34:09.908 DEBUG qcprludev10 cbinds and rbinds
## 2023 10 06 12:34:09.908 DEBUG qcprludev10 Object@DF1batch= RNASeqV2
## 2023 10 06 12:34:09.908 DEBUG qcprludev10 Object@DF2batch= Agilent4502
## 2023 10 06 12:34:09.909 DEBUG qcprludev10 row.names(sampBatch)
## 2023 10 06 12:34:09.909 DEBUG qcprludev10 TCGA-A1-AOSH-01A-11R-A084-07.RNASeqV2 TCGA-A1-AOSJ-01A-11R
## 2023 10 06 12:34:09.910 DEBUG qcprludev10 colnames(sampBatch)
## 2023 10 06 12:34:09.910 DEBUG qcprludev10 sample batch
## 2023 10 06 12:34:09.911 DEBUG qcprludev10 inside train, after getData4EB
## 2023 10 06 12:34:09.911 DEBUG qcprludev10 call EB MBatch

```

```

## 2023 10 06 12:34:09.911 DEBUG qcprludev10 TDC HERE trainEB mbatch
## 2023 10 06 12:34:09.912 DEBUG qcprludev10 EBNplus args Obj
## 2023 10 06 12:34:09.912 DEBUG qcprludev10 EBNplus args par.prior
## 2023 10 06 12:34:09.912 DEBUG qcprludev10 EBNplus args minSampleNum
## 2023 10 06 12:34:09.913 DEBUG qcprludev10 EBNplus theEBNP_PriorPlotsFile= /BEA/BatchEffectsPackage_d
## 2023 10 06 12:34:09.913 DEBUG qcprludev10 dat
## [1] 7785 576
## 2023 10 06 12:34:09.913 DEBUG qcprludev10 saminfo
## [1] 576 2
## 2023 10 06 12:34:09.914 DEBUG qcprludev10 check column and row names
## 2023 10 06 12:34:10.028 DEBUG qcprludev10 in design.mat_plus
## 2023 10 06 12:34:10.029 DEBUG qcprludev10 in build.design_plus
## 2023 10 06 12:34:10.030 DEBUG qcprludev10 after build.design_plus loop
## 2023 10 06 12:34:10.030 DEBUG qcprludev10 design
## 2023 10 06 12:34:10.031 DEBUG qcprludev10 list.batch_plus(saminfo)
## 2023 10 06 12:34:10.031 DEBUG qcprludev10 in list.batch_plus
## 2023 10 06 12:34:10.032 DEBUG qcprludev10 list.batch_plus tmp1
## 2023 10 06 12:34:10.032 DEBUG qcprludev10 list.batch_plus uniTmp[i]
## 2023 10 06 12:34:10.032 DEBUG qcprludev10 list.batch_plus uniTmp[i]
## 2023 10 06 12:34:10.033 DEBUG qcprludev10 list.batch_plus batches
## 2023 10 06 12:34:10.033 DEBUG qcprludev10 after list.batch_plus(saminfo)
## 2023 10 06 12:34:10.034 DEBUG qcprludev10 n.batches
## 2023 10 06 12:34:10.034 DEBUG qcprludev10 n.array
## 2023 10 06 12:34:11.738 DEBUG qcprludev10 missbatch matrix
## 2023 10 06 12:34:11.764 DEBUG qcprludev10 There are 46 genes that were removed because of whole ba
## 2023 10 06 12:34:12.433 DEBUG qcprludev10 B.hat
## 2023 10 06 12:34:12.436 DEBUG qcprludev10 grand.mean
## 2023 10 06 12:34:12.729 DEBUG qcprludev10 var.pooled
## 2023 10 06 12:34:12.732 DEBUG qcprludev10 stand.mean 1
## 2023 10 06 12:34:12.777 DEBUG qcprludev10 stand.mean 2
## 2023 10 06 12:34:12.807 DEBUG qcprludev10 s.data
## 2023 10 06 12:34:13.467 DEBUG qcprludev10 gamma.hat
## 2023 10 06 12:34:13.838 DEBUG qcprludev10 plot priors
## 2023 10 06 12:34:13.838 DEBUG qcprludev10 plotPrior priorPlotsFile= /BEA/BatchEffectsPackage_data/te
## 2023 10 06 12:34:13.994 DEBUG qcprludev10 plotPrior - else
## 2023 10 06 12:34:14.131 DEBUG qcprludev10 plotPrior - finished prior plots
## 2023 10 06 12:34:14.220 DEBUG qcprludev10 parametric adjustments
## 2023 10 06 12:34:14.753 DEBUG qcprludev10 parametric adjustments
## 2023 10 06 12:34:15.039 DEBUG qcprludev10 in getBayesData
## 2023 10 06 12:34:15.040 DEBUG qcprludev10 dim(s.data)=7739 576
## 2023 10 06 12:34:15.040 DEBUG qcprludev10 dim(stand.mean)=7739 576
## 2023 10 06 12:34:15.326 DEBUG qcprludev10 dim(bayesdata)=7739 576
## 2023 10 06 12:34:15.356 DEBUG qcprludev10 Second dim(bayesdata)=7739 576
## 2023 10 06 12:34:15.356 DEBUG qcprludev10 keep the dim of Original.dat, keep the gene with all NA
## 2023 10 06 12:34:15.377 DEBUG qcprludev10 resultsDat
## 2023 10 06 12:34:15.378 DEBUG qcprludev10 after EB MBatch
## 2023 10 06 12:34:15.379 DEBUG qcprludev10 inside TEST branch MBatch
## 2023 10 06 12:34:15.383 DEBUG qcprludev10 test
## 2023 10 06 12:34:15.383 DEBUG qcprludev10 dim(mat1)=7785 123
## 2023 10 06 12:34:15.384 DEBUG qcprludev10 dim(mat1)=
## 2023 10 06 12:34:15.384 DEBUG qcprludev10 dim(mat2)=7785 123
## 2023 10 06 12:34:15.384 DEBUG qcprludev10 dim(mat2)=
## 2023 10 06 12:34:15.453 DEBUG qcprludev10 dim(dat)=7785 246

```

```

## 2023 10 06 12:34:15.453 DEBUG qcprludev10 dim(dat)=
## 2023 10 06 12:34:15.462 DEBUG qcprludev10 after parlist
## 2023 10 06 12:34:15.463 DEBUG qcprludev10 dim(dat)=7785 246
## 2023 10 06 12:34:15.463 DEBUG qcprludev10 valid.genes
## 2023 10 06 12:34:15.464 DEBUG qcprludev10 nonValidGenes
## 2023 10 06 12:34:15.470 DEBUG qcprludev10 Non-valid genes were removed before adjustment
## 2023 10 06 12:34:15.470 DEBUG qcprludev10 after Non-valid
## 2023 10 06 12:34:15.471 DEBUG qcprludev10 dim(dat)=7739 246
## 2023 10 06 12:34:15.506 DEBUG qcprludev10 after getStandData
## 2023 10 06 12:34:15.507 DEBUG qcprludev10 dim(stand.data)=7739 246
## 2023 10 06 12:34:15.508 DEBUG qcprludev10 batches
## 2023 10 06 12:34:15.508 DEBUG qcprludev10 after batch.design
## 2023 10 06 12:34:15.509 DEBUG qcprludev10 dim(batch.design)=246 2
## 2023 10 06 12:34:15.509 DEBUG qcprludev10 n.batches
## 2023 10 06 12:34:15.509 DEBUG qcprludev10 in getBayesData
## 2023 10 06 12:34:15.510 DEBUG qcprludev10 dim(s.data)=7739 246
## 2023 10 06 12:34:15.510 DEBUG qcprludev10 dim(stand.mean)=7739 246
## 2023 10 06 12:34:15.547 DEBUG qcprludev10 dim(bayesdata)=7739 246
## 2023 10 06 12:34:15.564 DEBUG qcprludev10 Second dim(bayesdata)=7739 246
## 2023 10 06 12:34:15.565 DEBUG qcprludev10 after bayesData
## 2023 10 06 12:34:15.565 DEBUG qcprludev10 dim(bayesData)=7739 246
## 2023 10 06 12:34:15.566 DEBUG qcprludev10 EBadj datNonValid
## 2023 10 06 12:34:15.580 DEBUG qcprludev10 EBadj EBadj after non valid
## 2023 10 06 12:34:15.581 DEBUG qcprludev10 Check the adjusted test set
## 2023 10 06 12:34:15.581 DEBUG qcprludev10 after train
## 2023 10 06 12:34:15.581 DEBUG qcprludev10 EBadj
## 2023 10 06 12:34:15.582 DEBUG qcprludev10 EBadj validation for validation results
## 2023 10 06 12:34:15.651 DEBUG qcprludev10 validation
## 2023 10 06 12:34:15.656 DEBUG qcprludev10 dim(mat1)=7785 175
## 2023 10 06 12:34:15.656 DEBUG qcprludev10 dim(mat1)=
## 2023 10 06 12:34:15.657 DEBUG qcprludev10 dim(mat2)=7785 175
## 2023 10 06 12:34:15.657 DEBUG qcprludev10 dim(mat2)=
## 2023 10 06 12:34:15.746 DEBUG qcprludev10 dim(dat)=7785 350
## 2023 10 06 12:34:15.746 DEBUG qcprludev10 dim(dat)=
## 2023 10 06 12:34:15.758 DEBUG qcprludev10 after parlist
## 2023 10 06 12:34:15.758 DEBUG qcprludev10 dim(dat)=7785 350
## 2023 10 06 12:34:15.759 DEBUG qcprludev10 valid.genes
## 2023 10 06 12:34:15.760 DEBUG qcprludev10 nonValidGenes
## 2023 10 06 12:34:15.768 DEBUG qcprludev10 Non-valid genes were removed before adjustment
## 2023 10 06 12:34:15.768 DEBUG qcprludev10 after Non-valid
## 2023 10 06 12:34:15.769 DEBUG qcprludev10 dim(dat)=7739 350
## 2023 10 06 12:34:15.988 DEBUG qcprludev10 after getStandData
## 2023 10 06 12:34:15.989 DEBUG qcprludev10 dim(stand.data)=7739 350
## 2023 10 06 12:34:15.989 DEBUG qcprludev10 batches
## 2023 10 06 12:34:15.990 DEBUG qcprludev10 after batch.design
## 2023 10 06 12:34:15.990 DEBUG qcprludev10 dim(batch.design)=350 2
## 2023 10 06 12:34:15.991 DEBUG qcprludev10 n.batches
## 2023 10 06 12:34:15.991 DEBUG qcprludev10 in getBayesData
## 2023 10 06 12:34:15.992 DEBUG qcprludev10 dim(s.data)=7739 350
## 2023 10 06 12:34:15.992 DEBUG qcprludev10 dim(stand.mean)=7739 350
## 2023 10 06 12:34:16.043 DEBUG qcprludev10 dim(bayesdata)=7739 350
## 2023 10 06 12:34:16.061 DEBUG qcprludev10 Second dim(bayesdata)=7739 350
## 2023 10 06 12:34:16.062 DEBUG qcprludev10 after bayesData
## 2023 10 06 12:34:16.062 DEBUG qcprludev10 dim(bayesData)=7739 350

```



```

## 2023 10 06 12:34:16.063 DEBUG qcprludev10 EBadj datNonValid
## 2023 10 06 12:34:16.088 DEBUG qcprludev10 EBadj EBadj after non valid
## 2023 10 06 12:34:16.088 DEBUG qcprludev10 objafterEB@validationEB
## 2023 10 06 12:34:16.089 DEBUG qcprludev10 EBadj train for training results
## 2023 10 06 12:34:16.089 DEBUG qcprludev10 train
## 2023 10 06 12:34:16.090 DEBUG qcprludev10 dim(mat1)=7785 288
## 2023 10 06 12:34:16.090 DEBUG qcprludev10 dim(mat1)=
## 2023 10 06 12:34:16.091 DEBUG qcprludev10 dim(mat2)=7785 288
## 2023 10 06 12:34:16.091 DEBUG qcprludev10 dim(mat2)=
## 2023 10 06 12:34:16.252 DEBUG qcprludev10 dim(dat)=7785 576
## 2023 10 06 12:34:16.253 DEBUG qcprludev10 dim(dat)=
## 2023 10 06 12:34:16.271 DEBUG qcprludev10 after parlist
## 2023 10 06 12:34:16.272 DEBUG qcprludev10 dim(dat)=7785 576
## 2023 10 06 12:34:16.272 DEBUG qcprludev10 valid.genes
## 2023 10 06 12:34:16.273 DEBUG qcprludev10 nonValidGenes
## 2023 10 06 12:34:16.286 DEBUG qcprludev10 Non-valid genes were removed before adjustment
## 2023 10 06 12:34:16.286 DEBUG qcprludev10 after Non-valid
## 2023 10 06 12:34:16.287 DEBUG qcprludev10 dim(dat)=7739 576
## 2023 10 06 12:34:16.367 DEBUG qcprludev10 after getStandData
## 2023 10 06 12:34:16.368 DEBUG qcprludev10 dim(stand.data)=7739 576
## 2023 10 06 12:34:16.368 DEBUG qcprludev10 batches
## 2023 10 06 12:34:16.369 DEBUG qcprludev10 after batch.design
## 2023 10 06 12:34:16.369 DEBUG qcprludev10 dim(batch.design)=576 2
## 2023 10 06 12:34:16.369 DEBUG qcprludev10 n.batches
## 2023 10 06 12:34:16.370 DEBUG qcprludev10 in getBayesData
## 2023 10 06 12:34:16.370 DEBUG qcprludev10 dim(s.data)=7739 576
## 2023 10 06 12:34:16.371 DEBUG qcprludev10 dim(stand.mean)=7739 576
## 2023 10 06 12:34:16.458 DEBUG qcprludev10 dim(bayesdata)=7739 576
## 2023 10 06 12:34:16.478 DEBUG qcprludev10 Second dim(bayesdata)=7739 576
## 2023 10 06 12:34:16.478 DEBUG qcprludev10 after bayesData
## 2023 10 06 12:34:16.479 DEBUG qcprludev10 dim(bayesData)=7739 576
## 2023 10 06 12:34:16.479 DEBUG qcprludev10 EBadj datNonValid
## 2023 10 06 12:34:16.521 DEBUG qcprludev10 EBadj EBadj after non valid
## 2023 10 06 12:34:16.521 DEBUG qcprludev10 trainingResults@trainEB
## 2023 10 06 12:34:16.522 DEBUG qcprludev10 EBadj train for test results
## 2023 10 06 12:34:16.522 DEBUG qcprludev10 test
## 2023 10 06 12:34:16.523 DEBUG qcprludev10 dim(mat1)=7785 123
## 2023 10 06 12:34:16.523 DEBUG qcprludev10 dim(mat1)=
## 2023 10 06 12:34:16.524 DEBUG qcprludev10 dim(mat2)=7785 123
## 2023 10 06 12:34:16.524 DEBUG qcprludev10 dim(mat2)=
## 2023 10 06 12:34:16.597 DEBUG qcprludev10 dim(dat)=7785 246
## 2023 10 06 12:34:16.597 DEBUG qcprludev10 dim(dat)=
## 2023 10 06 12:34:16.606 DEBUG qcprludev10 after parlist
## 2023 10 06 12:34:16.606 DEBUG qcprludev10 dim(dat)=7785 246
## 2023 10 06 12:34:16.607 DEBUG qcprludev10 valid.genes
## 2023 10 06 12:34:16.608 DEBUG qcprludev10 nonValidGenes
## 2023 10 06 12:34:16.614 DEBUG qcprludev10 Non-valid genes were removed before adjustment
## 2023 10 06 12:34:16.614 DEBUG qcprludev10 after Non-valid
## 2023 10 06 12:34:16.614 DEBUG qcprludev10 dim(dat)=7739 246
## 2023 10 06 12:34:16.652 DEBUG qcprludev10 after getStandData
## 2023 10 06 12:34:16.653 DEBUG qcprludev10 dim(stand.data)=7739 246
## 2023 10 06 12:34:16.653 DEBUG qcprludev10 batches
## 2023 10 06 12:34:16.654 DEBUG qcprludev10 after batch.design
## 2023 10 06 12:34:16.654 DEBUG qcprludev10 dim(batch.design)=246 2

```

```

## 2023 10 06 12:34:16.654 DEBUG qcprludev10 n.batches
## 2023 10 06 12:34:16.655 DEBUG qcprludev10 in getBayesData
## 2023 10 06 12:34:16.655 DEBUG qcprludev10 dim(s.data)=7739 246
## 2023 10 06 12:34:16.656 DEBUG qcprludev10 dim(stand.mean)=7739 246
## 2023 10 06 12:34:16.695 DEBUG qcprludev10 dim(bayesdata)=7739 246
## 2023 10 06 12:34:16.708 DEBUG qcprludev10 Second dim(bayesdata)=7739 246
## 2023 10 06 12:34:16.709 DEBUG qcprludev10 after bayesData
## 2023 10 06 12:34:16.710 DEBUG qcprludev10 dim(bayesData)=7739 246
## 2023 10 06 12:34:16.710 DEBUG qcprludev10 EBadj datNonValid
## 2023 10 06 12:34:16.726 DEBUG qcprludev10 EBadj EBadj after non valid
## 2023 10 06 12:34:16.727 DEBUG qcprludev10 testResults@testEB
## 2023 10 06 12:34:16.782 DEBUG qcprludev10 finishing BeaEBNPlus
## [1] "TestSet1"
## [1] TRUE
## [1] 7785 123
##      TCGA-A1-AOSE-01A-11R-A084-07 TCGA-A1-AOSP-01A-11R-A084-07
## A1BG                               7.5920                      5.210
## A2BP1                             -2.0160                      -2.016
## A2M                                13.9410                      12.357
## A2ML1                              0.3038                       9.738
##      TCGA-A2-A04Y-01A-21R-A034-07 TCGA-A2-A0CP-01A-11R-A034-07
## A1BG                               7.5074                      8.54000
## A2BP1                              1.5477                      -0.12437
## A2M                                13.5140                      14.51200
## A2ML1                             -2.0160                      -0.12437
## [1] "TestSet2"
## [1] TRUE
## [1] 7785 123
##      TCGA-A1-AOSE-01A-11R-A084-07 TCGA-A1-AOSP-01A-11R-A084-07
## A1BG                               0.2798333                 -1.291167
## A2BP1                             -1.5346667                 -1.133833
## A2M                                1.0385000                 -0.549500
## A2ML1                              0.3715000                 1.332000
##      TCGA-A2-A04Y-01A-21R-A034-07 TCGA-A2-A0CP-01A-11R-A034-07
## A1BG                               0.3046667                 0.9571667
## A2BP1                             -0.8158333                 -1.4495000
## A2M                                0.5885000                 1.2132500
## A2ML1                              0.6145000                 1.1390000
## [1] "TrainingSet1"
## [1] TRUE
## [1] 7785 288
##      TCGA-A1-AOSH-01A-11R-A084-07 TCGA-A1-AOSJ-01A-11R-A084-07
## A1BG                               7.4960                      7.820000
## A2BP1                             -0.7169                      -2.016000
## A2M                                13.7700                      14.432000
## A2ML1                              0.7585                      -0.030913
##      TCGA-A1-AOSK-01A-12R-A084-07 TCGA-A1-AOSM-01A-11R-A084-07
## A1BG                               6.440                      8.265
## A2BP1                             -2.016                      -0.829
## A2M                                12.032                      13.120
## A2ML1                              4.652                      -0.829
## [1] "TrainingSet2"
## [1] TRUE
## [1] 7785 288

```

```

##      TCGA-A1-AOSH-01A-11R-A084-07 TCGA-A1-AOSJ-01A-11R-A084-07
## A1BG      0.7486667      0.750000
## A2BP1     -1.3048333     -1.357833
## A2M       1.1605000      1.677750
## A2ML1     0.3575000      0.518500
##      TCGA-A1-AOSK-01A-12R-A084-07 TCGA-A1-AOSM-01A-11R-A084-07
## A1BG      -0.5471667      0.5976667
## A2BP1     -2.1845000     -1.1756667
## A2M       -0.8597500      0.5745000
## A2ML1     0.3060000      0.2635000
## [1] "TrainingResults"
## [1] TRUE
## [1] 7785  576
##      TCGA-A1-AOSH-01A-11R-A084-07.Agilent4502
## A1BG      4.2922928
## A2BP1     -0.3358343
## A2M       7.5542896
## A2ML1     1.4633094
##      TCGA-A1-AOSH-01A-11R-A084-07.RNASeqV2
## A1BG      4.02642841
## A2BP1     -1.30645343
## A2M       7.37907814
## A2ML1     0.06540906
##      TCGA-A1-AOSJ-01A-11R-A084-07.Agilent4502
## A1BG      4.2939693
## A2BP1     -0.4097362
## A2M       8.0769652
## A2ML1     1.9279318
##      TCGA-A1-AOSJ-01A-11R-A084-07.RNASeqV2
## A1BG      4.3031814
## A2BP1     -2.2213301
## A2M       8.0374491
## A2ML1     -0.4942765
## [1] "ValidationSet1"
## [1] TRUE
## [1] 7785  175
##      TCGA-A1-AOSD-01A-11R-A115-07 TCGA-A2-A04P-01A-31R-A034-07
## A1BG      7.1552      6.7681
## A2BP1     -0.7908     -2.0160
## A2M      14.5380     13.1800
## A2ML1     2.9738     10.6840
##      TCGA-A2-A04V-01A-21R-A034-07 TCGA-A2-A04W-01A-31R-A115-07
## A1BG      7.2050      7.185
## A2BP1     -2.0160     -2.016
## A2M      13.2340     14.490
## A2ML1     0.9083      6.947
## [1] "ValidationSet2"
## [1] TRUE
## [1] 7785  175
##      TCGA-A1-AOSD-01A-11R-A115-07 TCGA-A2-A04P-01A-31R-A034-07
## A1BG      0.9493333     -0.1568333
## A2BP1     0.5391667     -0.8194000
## A2M       0.2420000      0.2440000
## A2ML1     0.4235000      1.9145000

```

```

##          TCGA-A2-A04V-01A-21R-A034-07 TCGA-A2-A04W-01A-31R-A115-07
## A1BG          0.44500          -0.7923333
## A2BP1         -1.18800          -1.3916667
## A2M           0.17925           1.9417500
## A2ML1         0.74500           0.9930000
## [1] "ValidationResults"
## [1] TRUE
## [1] 7785 350
##          TCGA-A1-A0SD-01A-11R-A115-07.Agilent4502
## A1BG          4.544614
## A2BP1         2.235393
## A2M           6.626155
## A2ML1         1.653776
##          TCGA-A1-A0SD-01A-11R-A115-07.RNASeqV2
## A1BG          3.735325
## A2BP1         -1.358497
## A2M           8.142868
## A2ML1         1.636034
##          TCGA-A2-A04P-01A-31R-A034-07.Agilent4502
## A1BG          3.1537028
## A2BP1         0.3410418
## A2M           6.6281763
## A2ML1         5.9565837
##          TCGA-A2-A04P-01A-31R-A034-07.RNASeqV2
## A1BG          3.404674
## A2BP1         -2.221330
## A2M           6.792313
## A2ML1         7.102485
## [1] "CorrectedResults"
## [1] FALSE
## NULL
## NULL

```

8 Example Output

A list of matrices with the following names().

```

##TestSet1
a matrix of the test set from data set 1

##TestSet2
a matrix of the test set from data set 2

##TrainingSet1
a matrix of the training set from data set 1

##TrainingSet2
a matrix of the training set from data set 2

##TrainingResults
a matrix of the corrected training data

##ValidationSet1
a matrix of the validation set from data set 1

##ValidationSet2
a matrix of the validation set from data set 2

```

##ValidationResults a matrix of the corrected validation data

##CorrectedResults

If Validation Ratio is 0, the validation set values will be 0 x 0 matrices, but this list member will contain a matrix of corrected values, using all replicates.

This also generates the prior plots output file like this one.

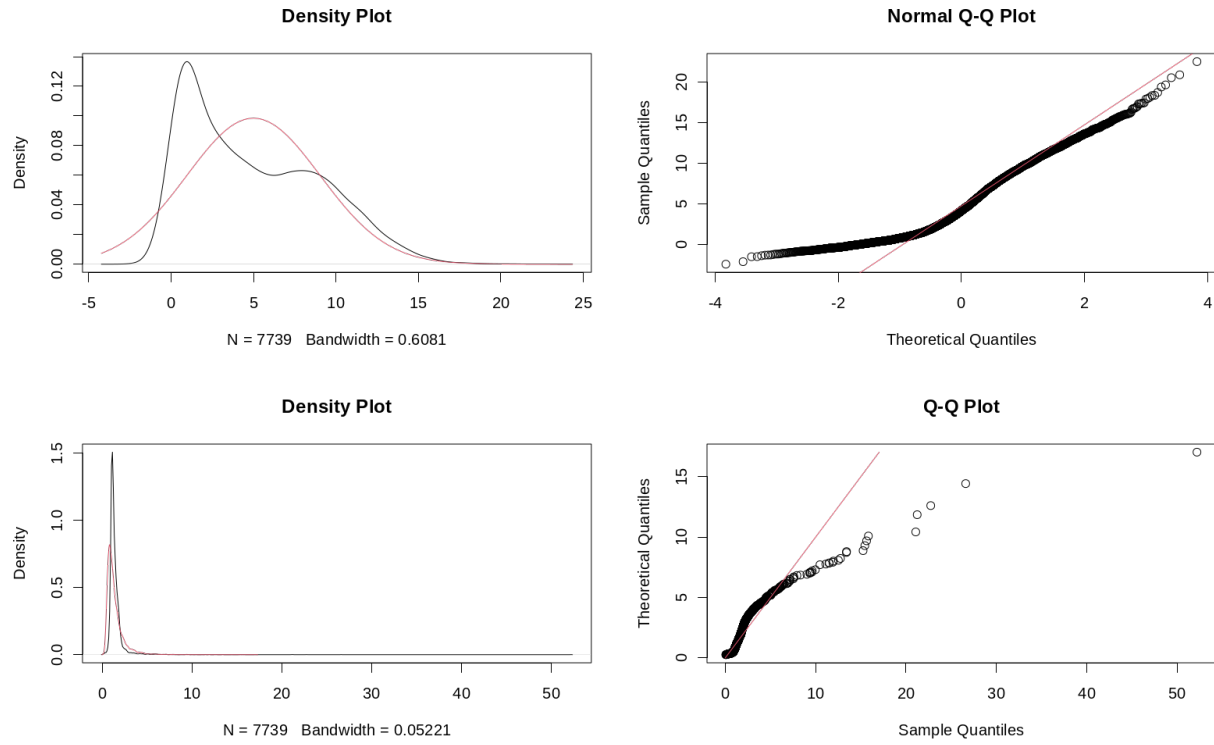


Figure 1: Prior Plots File