Using MBatch Corrections: EBNPlus_TrainAndValidateFromVector_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_TrainAndValidateFromVector_Structures creates and returns a training and validation set and corrected data, based on the training and validation set and provided vectors of samples to use for training and validation, instead of using replicates and percentages.

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_TrainAndValidateFromVector_Structures(theDataMatrix1, theDataMatrix2, theBatchId1, theBatchId2, theEBNP_PsuedoReplicates1Train, theEBNP_PsuedoReplicates2Train, theEBNP_PsuedoReplicates1Validation, theEBNP_PsuedoReplicates2Validation, theEBNP_BatchWithZero, theEBNP_FixDataSet, theEBNP_CorrectForZero, theEBNP_ParametricPriorsFlag, theEBNP_TestRatio=0, theSeed=NULL, theTestSeed=NULL, theEBNP_PriorPlotsFile = NULL, theEBNP_MinSampleNum = 3)

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_PsuedoReplicates1Train
A vector of sample ids to use as replicates for data set 1 for training.

##theEBNP_PsuedoReplicates2Train
A vector of sample ids to use as replicates for data set 2 for training.

##theEBNP_PsuedoReplicates1Validation
A vector of sample ids to use as replicates for data set 1 for validation. Use NULL if no validation is desired. This means corrections will be done.

##theEBNP_PsuedoReplicates2Validation A vector of sample ids to use as replicates for data set 2 for validation. Use NULL if no validation is desired. This means corrections will be done.

##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)

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

##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.

##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_TestRatio If you are doing corrections, this should be set to 0. Any other number is doing training and validation tests, to allow viewing of internal diagnostics.

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_Structures2")
  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)
  {
    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)
}


printMatrix <- function(theMatrix)
{
  print("printMatrix")
  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))
  # pseudoreplicate vectors
    theEBNP_PsuedoReplicates1Train <- c("TCGA-A1-A0SH-01A-11R-A084-07", "TCGA-A1-A0SJ-01A-11R-A084-07
  theEBNP_PsuedoReplicates2Train <- c("TCGA-A1-A0SH-01A-11R-A084-07", "TCGA-A1-A0SJ-01A-11R-A084-07",
    theEBNP_PsuedoReplicates1Validation <- c("TCGA-A1-A0SD-01A-11R-A115-07", "TCGA-A1-A0SK-01A-12R-A084
    theEBNP_PsuedoReplicates2Validation <- c("TCGA-A1-A0SD-01A-11R-A115-07", "TCGA-A1-A0SK-01A-12R-A084
    # call function
    print("EBNPlus_TrainAndValidateReplicates_Structures")
    resultsList <- EBNPlus_TrainAndValidateFromVector_Structures(theDataMatrix1, theDataMatrix2,
          theBatchId1, theBatchId2,
          theEBNP_PsuedoReplicates1Train,
          theEBNP_PsuedoReplicates2Train,
          theEBNP_PsuedoReplicates1Validation,
          theEBNP_PsuedoReplicates2Validation,
          theEBNP_BatchWithZero="1",
          theEBNP_FixDataSet=as.numeric(NA),
          theEBNP_CorrectForZero=TRUE,
          theEBNP_ParametricPriorsFlag=TRUE,
          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 12:34:24.299 DEBUG qcprludev10 EBNPlus_TrainAndValidateFromVector_Structures - start
## 2023 10 06 12:34:24.300 DEBUG qcprludev10 removeDuplicatesFromRows
## 2023 10 06 12:34:24.301 DEBUG qcprludev10 removeDuplicatesFromRows
## 2023 10 06 12:34:24.301 DEBUG qcprludev10 removeDuplicatesFromColumns
## 2023 10 06 12:34:24.302 DEBUG qcprludev10 removeDuplicatesFromColumns
## 2023 10 06 12:34:24.302 DEBUG qcprludev10 EBNPlus myPriorPlotsFile= /BEA/BatchEffectsPackage_data/te
## 2023 10 06 12:34:24.303 DEBUG qcprludev10 dim(theDataMatrix1)=10000 1215
## 2023 10 06 12:34:24.303 DEBUG qcprludev10 dim(theDataMatrix2)=10000 600
## 2023 10 06 12:34:24.303 DEBUG qcprludev10 remove unknown genes, that start with ?
## 2023 10 06 12:34:24.402 DEBUG qcprludev10 dim(theDataMatrix1)=10000 1215
## 2023 10 06 12:34:24.403 DEBUG qcprludev10 dim(theDataMatrix2)=10000 600
## 2023 10 06 12:34:24.403 DEBUG qcprludev10 make EBNplus
## 2023 10 06 12:34:24.407 DEBUG qcprludev10 before callNextMethod
## 2023 10 06 12:34:24.408 DEBUG qcprludev10 dim(mData1)=10000 1215
## 2023 10 06 12:34:24.408 DEBUG qcprludev10 dim(mData2)=10000 600
## 2023 10 06 12:34:24.409 DEBUG qcprludev10 after callNextMethod
## 2023 10 06 12:34:24.409 DEBUG qcprludev10 dim(mData1)=10000 1215
## 2023 10 06 12:34:24.410 DEBUG qcprludev10 dim(mData2)=10000 600
## 2023 10 06 12:34:24.410 DEBUG qcprludev10 after EBNplus
## 2023 10 06 12:34:24.410 DEBUG qcprludev10 dim(ebObj@mData1)=10000 1215
## 2023 10 06 12:34:24.411 DEBUG qcprludev10 dim(ebObj@mData2)=10000 600
## 2023 10 06 12:34:24.413 DEBUG qcprludev10 getBiComOrder
## 2023 10 06 12:34:24.413 DEBUG qcprludev10 makeCommonRows
## 2023 10 06 12:34:24.477 DEBUG qcprludev10 dim(mData1)=7785 1215
## 2023 10 06 12:34:24.478 DEBUG qcprludev10 dim(mData2)=7785 600
## 2023 10 06 12:34:24.478 DEBUG qcprludev10 makeCommonCols
## 2023 10 06 12:34:24.508 DEBUG qcprludev10 dim(mData1)=7785 586
## 2023 10 06 12:34:24.508 DEBUG qcprludev10 dim(mData2)=7785 586
## 2023 10 06 12:34:24.509 DEBUG qcprludev10 asSameOrder
## 2023 10 06 12:34:24.527 DEBUG qcprludev10 asSameOrder before
## 2023 10 06 12:34:24.528 DEBUG qcprludev10 asSameOrder after if 1
## 2023 10 06 12:34:24.546 DEBUG qcprludev10 asSameOrder after if 2
## 2023 10 06 12:34:24.546 DEBUG qcprludev10 row.names(mat1)
## 2023 10 06 12:34:24.547 DEBUG qcprludev10 rownames(mat1)
## 2023 10 06 12:34:24.547 DEBUG qcprludev10 row.names(mat2)
## 2023 10 06 12:34:24.547 DEBUG qcprludev10 rownames(mat2)
## 2023 10 06 12:34:24.548 DEBUG qcprludev10 m.i
## 2023 10 06 12:34:24.566 DEBUG qcprludev10 asSameOrder after if 3
## 2023 10 06 12:34:24.567 DEBUG qcprludev10 dim(ebObj@mat1Com)=7785 586
## 2023 10 06 12:34:24.567 DEBUG qcprludev10 dim(ebObj@mat2Com)=7785 586
## 2023 10 06 12:34:24.567 DEBUG qcprludev10 get Validation and Training sets from vectors
## 2023 10 06 12:34:24.591 DEBUG qcprludev10 dim(ebObj@mat1Validation)=10000 175
## 2023 10 06 12:34:24.592 DEBUG qcprludev10 dim(ebObj@mat2Validation)=10000 175
## 2023 10 06 12:34:24.592 DEBUG qcprludev10 dim(ebObj@mat1Train)=10000 288
## 2023 10 06 12:34:24.593 DEBUG qcprludev10 dim(ebObj@mat2Train)=10000 288
## 2023 10 06 12:34:24.593 DEBUG qcprludev10 before Test Set
## 2023 10 06 12:34:24.625 DEBUG qcprludev10 after Test Set
## 2023 10 06 12:34:24.668 DEBUG qcprludev10 train
## 2023 10 06 12:34:24.685 DEBUG qcprludev10 inside train
## 2023 10 06 12:34:24.686 DEBUG qcprludev10 train theEBNP_PriorPlotsFile= /BEA/BatchEffectsPackage_data
## 2023 10 06 12:34:24.703 DEBUG qcprludev10 colnames(Object@mat1Train)=TCGA-A1-A0SJ-01A-11R-A084-07.RNA
## 2023 10 06 12:34:24.703 DEBUG qcprludev10 colnames(Object@mat2Train)=TCGA-A1-A0SJ-01A-11R-A084-07.Agi
## 2023 10 06 12:34:24.704 DEBUG qcprludev10 inside train, call getData4EB
## 2023 10 06 12:34:24.901 DEBUG qcprludev10 data4EB <- as.matrix
## 2023 10 06 12:34:24.901 DEBUG qcprludev10 dim(data4EB)=7785 404
## 2023 10 06 12:34:24.902 DEBUG qcprludev10 cbinds and rbinds
## 2023 10 06 12:34:24.902 DEBUG qcprludev10 Object@DF1batch= RNASeqV2
## 2023 10 06 12:34:24.903 DEBUG qcprludev10 Object@DF2batch= Agilent4502
## 2023 10 06 12:34:24.904 DEBUG qcprludev10 row.names(sampBatch)
## 2023 10 06 12:34:24.904 DEBUG qcprludev10 TCGA-A1-A0SJ-01A-11R-A084-07.RNASeqV2 TCGA-A2-A04N-01A-11R-
## 2023 10 06 12:34:24.904 DEBUG qcprludev10 colnames(sampBatch)
## 2023 10 06 12:34:24.905 DEBUG qcprludev10 sample batch
## 2023 10 06 12:34:24.905 DEBUG qcprludev10 inside train, after getData4EB
## 2023 10 06 12:34:24.906 DEBUG qcprludev10 call EB MBatch
## 2023 10 06 12:34:24.981 DEBUG qcprludev10 TDC HERE trainEB mbatch
## 2023 10 06 12:34:24.981 DEBUG qcprludev10 EBNplus args Obj
## 2023 10 06 12:34:24.982 DEBUG qcprludev10 EBNplus args par.prior
## 2023 10 06 12:34:24.982 DEBUG qcprludev10 EBNplus args minSampleNum
## 2023 10 06 12:34:24.983 DEBUG qcprludev10 EBNplus theEBNP_PriorPlotsFile= /BEA/BatchEffectsPackage_da
## 2023 10 06 12:34:24.983 DEBUG qcprludev10 dat
## [1] 7785  404
## 2023 10 06 12:34:24.984 DEBUG qcprludev10 saminfo
## [1] 404   2
## 2023 10 06 12:34:24.984 DEBUG qcprludev10 check column and row names
## 2023 10 06 12:34:25.064 DEBUG qcprludev10 in design.mat_plus
## 2023 10 06 12:34:25.065 DEBUG qcprludev10 in build.design_plus
## 2023 10 06 12:34:25.065 DEBUG qcprludev10 after build.design_plus loop
## 2023 10 06 12:34:25.066 DEBUG qcprludev10 design
## 2023 10 06 12:34:25.066 DEBUG qcprludev10 list.batch_plus(saminfo)
## 2023 10 06 12:34:25.067 DEBUG qcprludev10 in list.batch_plus
## 2023 10 06 12:34:25.067 DEBUG qcprludev10 list.batch_plus tmp1
## 2023 10 06 12:34:25.068 DEBUG qcprludev10 list.batch_plus uniTmp[i]
## 2023 10 06 12:34:25.068 DEBUG qcprludev10 list.batch_plus uniTmp[i]
## 2023 10 06 12:34:25.069 DEBUG qcprludev10 list.batch_plus batches
## 2023 10 06 12:34:25.069 DEBUG qcprludev10 after list.batch_plus(saminfo)
## 2023 10 06 12:34:25.069 DEBUG qcprludev10 n.batches
## 2023 10 06 12:34:25.070 DEBUG qcprludev10 n.array
## 2023 10 06 12:34:26.598 DEBUG qcprludev10 missbatch matrix
## 2023 10 06 12:34:26.612 DEBUG qcprludev10 There are  63  genes that were removed because of whole ba
## 2023 10 06 12:34:27.122 DEBUG qcprludev10 B.hat
## 2023 10 06 12:34:27.123 DEBUG qcprludev10 grand.mean
## 2023 10 06 12:34:27.339 DEBUG qcprludev10 var.pooled
## 2023 10 06 12:34:27.342 DEBUG qcprludev10 stand.mean 1
## 2023 10 06 12:34:27.372 DEBUG qcprludev10 stand.mean 2
## 2023 10 06 12:34:27.393 DEBUG qcprludev10 s.data
## 2023 10 06 12:34:27.910 DEBUG qcprludev10 gamma.hat
## 2023 10 06 12:34:28.240 DEBUG qcprludev10 plot priors
## 2023 10 06 12:34:28.241 DEBUG qcprludev10 plotPrior priorPlotsFile= /BEA/BatchEffectsPackage_data/tes

## 2023 10 06 12:34:28.397 DEBUG qcprludev10 plotPrior - else

## 2023 10 06 12:34:28.536 DEBUG qcprludev10 plotPrior - finished prior plots
## 2023 10 06 12:34:28.626 DEBUG qcprludev10 parametric adjustments
## 2023 10 06 12:34:28.944 DEBUG qcprludev10 parametric adjustments
## 2023 10 06 12:34:29.164 DEBUG qcprludev10 in getBayesData
## 2023 10 06 12:34:29.165 DEBUG qcprludev10 dim(s.data)=7722 404
## 2023 10 06 12:34:29.165 DEBUG qcprludev10 dim(stand.mean)=7722 404
## 2023 10 06 12:34:29.439 DEBUG qcprludev10 dim(bayesdata)=7722 404
## 2023 10 06 12:34:29.453 DEBUG qcprludev10 Second dim(bayesdata)=7722 404
## 2023 10 06 12:34:29.454 DEBUG qcprludev10 keep the dim of Original.dat, keep the gene with all NA
## 2023 10 06 12:34:29.464 DEBUG qcprludev10 resultsDat
## 2023 10 06 12:34:29.465 DEBUG qcprludev10 after EB MBatch
## 2023 10 06 12:34:29.466 DEBUG qcprludev10 inside TEST branch MBatch
## 2023 10 06 12:34:29.469 DEBUG qcprludev10 test
## 2023 10 06 12:34:29.470 DEBUG qcprludev10 dim(mat1)=10000 86
## 2023 10 06 12:34:29.470 DEBUG qcprludev10 dim(mat1)=
## 2023 10 06 12:34:29.470 DEBUG qcprludev10 dim(mat2)=10000 86
## 2023 10 06 12:34:29.471 DEBUG qcprludev10 dim(mat2)=
## 2023 10 06 12:34:29.529 DEBUG qcprludev10 dim(dat)=7785 172
## 2023 10 06 12:34:29.530 DEBUG qcprludev10 dim(dat)=
## 2023 10 06 12:34:29.536 DEBUG qcprludev10 after parlist
## 2023 10 06 12:34:29.537 DEBUG qcprludev10 dim(dat)=7785 172
## 2023 10 06 12:34:29.537 DEBUG qcprludev10 valid.genes
## 2023 10 06 12:34:29.538 DEBUG qcprludev10 nonValidGenes
## 2023 10 06 12:34:29.542 DEBUG qcprludev10 Non-valid genes were removed before adjustment
## 2023 10 06 12:34:29.543 DEBUG qcprludev10 after Non-valid
## 2023 10 06 12:34:29.543 DEBUG qcprludev10 dim(dat)=7722 172
## 2023 10 06 12:34:29.563 DEBUG qcprludev10 after getStandData
## 2023 10 06 12:34:29.564 DEBUG qcprludev10 dim(stand.data)=7722 172
## 2023 10 06 12:34:29.564 DEBUG qcprludev10 batches
## 2023 10 06 12:34:29.565 DEBUG qcprludev10 after batch.design
## 2023 10 06 12:34:29.565 DEBUG qcprludev10 dim(batch.design)=172 2
## 2023 10 06 12:34:29.565 DEBUG qcprludev10 n.batches
## 2023 10 06 12:34:29.566 DEBUG qcprludev10 in getBayesData
## 2023 10 06 12:34:29.566 DEBUG qcprludev10 dim(s.data)=7722 172
## 2023 10 06 12:34:29.567 DEBUG qcprludev10 dim(stand.mean)=7722 172
## 2023 10 06 12:34:29.586 DEBUG qcprludev10 dim(bayesdata)=7722 172
## 2023 10 06 12:34:29.593 DEBUG qcprludev10 Second dim(bayesdata)=7722 172
## 2023 10 06 12:34:29.593 DEBUG qcprludev10 after bayesData
## 2023 10 06 12:34:29.594 DEBUG qcprludev10 dim(bayesData)=7722 172
## 2023 10 06 12:34:29.594 DEBUG qcprludev10 EBadj datNonValid
## 2023 10 06 12:34:29.603 DEBUG qcprludev10 EBadj EBadj after non valid
## 2023 10 06 12:34:29.604 DEBUG qcprludev10 Check the adjusted test set
## 2023 10 06 12:34:29.604 DEBUG qcprludev10 after train
## 2023 10 06 12:34:29.604 DEBUG qcprludev10 EBadj
## 2023 10 06 12:34:29.605 DEBUG qcprludev10 EBadj train
## 2023 10 06 12:34:29.605 DEBUG qcprludev10 train
## 2023 10 06 12:34:29.606 DEBUG qcprludev10 dim(mat1)=10000 202
## 2023 10 06 12:34:29.606 DEBUG qcprludev10 dim(mat1)=
## 2023 10 06 12:34:29.606 DEBUG qcprludev10 dim(mat2)=10000 202
## 2023 10 06 12:34:29.607 DEBUG qcprludev10 dim(mat2)=
## 2023 10 06 12:34:29.728 DEBUG qcprludev10 dim(dat)=7785 404
## 2023 10 06 12:34:29.729 DEBUG qcprludev10 dim(dat)=
## 2023 10 06 12:34:29.743 DEBUG qcprludev10 after parlist
## 2023 10 06 12:34:29.743 DEBUG qcprludev10 dim(dat)=7785 404
## 2023 10 06 12:34:29.744 DEBUG qcprludev10 valid.genes
## 2023 10 06 12:34:29.745 DEBUG qcprludev10 nonValidGenes
## 2023 10 06 12:34:29.754 DEBUG qcprludev10 Non-valid genes were removed before adjustment
## 2023 10 06 12:34:29.754 DEBUG qcprludev10 after Non-valid
## 2023 10 06 12:34:29.754 DEBUG qcprludev10 dim(dat)=7722 404
## 2023 10 06 12:34:29.806 DEBUG qcprludev10 after getStandData
## 2023 10 06 12:34:29.806 DEBUG qcprludev10 dim(stand.data)=7722 404
## 2023 10 06 12:34:29.807 DEBUG qcprludev10 batches
## 2023 10 06 12:34:29.807 DEBUG qcprludev10 after batch.design
## 2023 10 06 12:34:29.808 DEBUG qcprludev10 dim(batch.design)=404 2
## 2023 10 06 12:34:29.808 DEBUG qcprludev10 n.batches
## 2023 10 06 12:34:29.809 DEBUG qcprludev10 in getBayesData
## 2023 10 06 12:34:29.809 DEBUG qcprludev10 dim(s.data)=7722 404
## 2023 10 06 12:34:29.809 DEBUG qcprludev10 dim(stand.mean)=7722 404
## 2023 10 06 12:34:29.853 DEBUG qcprludev10 dim(bayesdata)=7722 404
## 2023 10 06 12:34:29.875 DEBUG qcprludev10 Second dim(bayesdata)=7722 404
## 2023 10 06 12:34:29.875 DEBUG qcprludev10 after bayesData
## 2023 10 06 12:34:29.876 DEBUG qcprludev10 dim(bayesData)=7722 404
## 2023 10 06 12:34:29.876 DEBUG qcprludev10 EBadj datNonValid
## 2023 10 06 12:34:29.906 DEBUG qcprludev10 EBadj EBadj after non valid
## 2023 10 06 12:34:29.907 DEBUG qcprludev10 objafterEB@trainEB
## 2023 10 06 12:34:29.907 DEBUG qcprludev10 EBadj validation
## 2023 10 06 12:34:29.908 DEBUG qcprludev10 validation
## 2023 10 06 12:34:29.917 DEBUG qcprludev10 dim(mat1)=10000 175
## 2023 10 06 12:34:29.917 DEBUG qcprludev10 dim(mat1)=
## 2023 10 06 12:34:29.918 DEBUG qcprludev10 dim(mat2)=10000 175
## 2023 10 06 12:34:29.918 DEBUG qcprludev10 dim(mat2)=
## 2023 10 06 12:34:30.034 DEBUG qcprludev10 dim(dat)=7785 350
## 2023 10 06 12:34:30.035 DEBUG qcprludev10 dim(dat)=
## 2023 10 06 12:34:30.047 DEBUG qcprludev10 after parlist
## 2023 10 06 12:34:30.048 DEBUG qcprludev10 dim(dat)=7785 350
## 2023 10 06 12:34:30.048 DEBUG qcprludev10 valid.genes
## 2023 10 06 12:34:30.049 DEBUG qcprludev10 nonValidGenes
## 2023 10 06 12:34:30.057 DEBUG qcprludev10 Non-valid genes were removed before adjustment
## 2023 10 06 12:34:30.057 DEBUG qcprludev10 after Non-valid
## 2023 10 06 12:34:30.058 DEBUG qcprludev10 dim(dat)=7722 350
## 2023 10 06 12:34:30.100 DEBUG qcprludev10 after getStandData
## 2023 10 06 12:34:30.102 DEBUG qcprludev10 dim(stand.data)=7722 350
## 2023 10 06 12:34:30.104 DEBUG qcprludev10 batches
## 2023 10 06 12:34:30.105 DEBUG qcprludev10 after batch.design
## 2023 10 06 12:34:30.107 DEBUG qcprludev10 dim(batch.design)=350 2
## 2023 10 06 12:34:30.107 DEBUG qcprludev10 n.batches
## 2023 10 06 12:34:30.109 DEBUG qcprludev10 in getBayesData
## 2023 10 06 12:34:30.109 DEBUG qcprludev10 dim(s.data)=7722 350
## 2023 10 06 12:34:30.110 DEBUG qcprludev10 dim(stand.mean)=7722 350
## 2023 10 06 12:34:30.168 DEBUG qcprludev10 dim(bayesdata)=7722 350
## 2023 10 06 12:34:30.184 DEBUG qcprludev10 Second dim(bayesdata)=7722 350
## 2023 10 06 12:34:30.185 DEBUG qcprludev10 after bayesData
## 2023 10 06 12:34:30.186 DEBUG qcprludev10 dim(bayesData)=7722 350
## 2023 10 06 12:34:30.186 DEBUG qcprludev10 EBadj datNonValid
## 2023 10 06 12:34:30.209 DEBUG qcprludev10 EBadj EBadj after non valid
## 2023 10 06 12:34:30.209 DEBUG qcprludev10 objafterEB@validationEB
## 2023 10 06 12:34:30.210 DEBUG qcprludev10 EBadj train for test results
## 2023 10 06 12:34:30.210 DEBUG qcprludev10 test
## 2023 10 06 12:34:30.211 DEBUG qcprludev10 dim(mat1)=10000 86
## 2023 10 06 12:34:30.211 DEBUG qcprludev10 dim(mat1)=
## 2023 10 06 12:34:30.211 DEBUG qcprludev10 dim(mat2)=10000 86
## 2023 10 06 12:34:30.212 DEBUG qcprludev10 dim(mat2)=
## 2023 10 06 12:34:30.276 DEBUG qcprludev10 dim(dat)=7785 172
## 2023 10 06 12:34:30.276 DEBUG qcprludev10 dim(dat)=
## 2023 10 06 12:34:30.283 DEBUG qcprludev10 after parlist
## 2023 10 06 12:34:30.283 DEBUG qcprludev10 dim(dat)=7785 172
## 2023 10 06 12:34:30.284 DEBUG qcprludev10 valid.genes
## 2023 10 06 12:34:30.285 DEBUG qcprludev10 nonValidGenes
## 2023 10 06 12:34:30.289 DEBUG qcprludev10 Non-valid genes were removed before adjustment
## 2023 10 06 12:34:30.290 DEBUG qcprludev10 after Non-valid
## 2023 10 06 12:34:30.290 DEBUG qcprludev10 dim(dat)=7722 172
## 2023 10 06 12:34:30.312 DEBUG qcprludev10 after getStandData
## 2023 10 06 12:34:30.313 DEBUG qcprludev10 dim(stand.data)=7722 172
## 2023 10 06 12:34:30.313 DEBUG qcprludev10 batches
## 2023 10 06 12:34:30.314 DEBUG qcprludev10 after batch.design
## 2023 10 06 12:34:30.314 DEBUG qcprludev10 dim(batch.design)=172 2
## 2023 10 06 12:34:30.315 DEBUG qcprludev10 n.batches
## 2023 10 06 12:34:30.315 DEBUG qcprludev10 in getBayesData
## 2023 10 06 12:34:30.315 DEBUG qcprludev10 dim(s.data)=7722 172
## 2023 10 06 12:34:30.316 DEBUG qcprludev10 dim(stand.mean)=7722 172
## 2023 10 06 12:34:30.343 DEBUG qcprludev10 dim(bayesdata)=7722 172
## 2023 10 06 12:34:30.352 DEBUG qcprludev10 Second dim(bayesdata)=7722 172
## 2023 10 06 12:34:30.353 DEBUG qcprludev10 after bayesData
## 2023 10 06 12:34:30.353 DEBUG qcprludev10 dim(bayesData)=7722 172
## 2023 10 06 12:34:30.354 DEBUG qcprludev10 EBadj datNonValid
## 2023 10 06 12:34:30.362 DEBUG qcprludev10 EBadj EBadj after non valid
## 2023 10 06 12:34:30.363 DEBUG qcprludev10 testResults@testEB
## [1] "TestSet1"
## [1] "printMatrix"
## [1] TRUE
## [1] 10000    86
## [1] "TestSet2"
## [1] "printMatrix"
## [1] TRUE
## [1] 10000    86
## [1] "TrainingSet1"
## [1] "printMatrix"
## [1] TRUE
## [1] 10000   202
## [1] "TrainingSet2"
## [1] "printMatrix"
## [1] TRUE
## [1] 10000   202
## [1] "TrainingResults"
## [1] "printMatrix"
## [1] TRUE
## [1] 7785  404
## [1] "ValidationSet1"
## [1] "printMatrix"
## [1] TRUE
## [1] 10000   175
## [1] "ValidationSet2"
## [1] "printMatrix"
## [1] TRUE
## [1] 10000   175
## [1] "ValidationResults"
## [1] "printMatrix"
## [1] TRUE
## [1] 7785  350
## [1] "CorrectedResults"
## [1] "printMatrix"
## [1] FALSE
## 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 corrections are done, this list member will contain a matrix of corrected values using all available replicates. In that case, the ValidationResults list member will be NULL or contain 0 x 0 matrices.

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

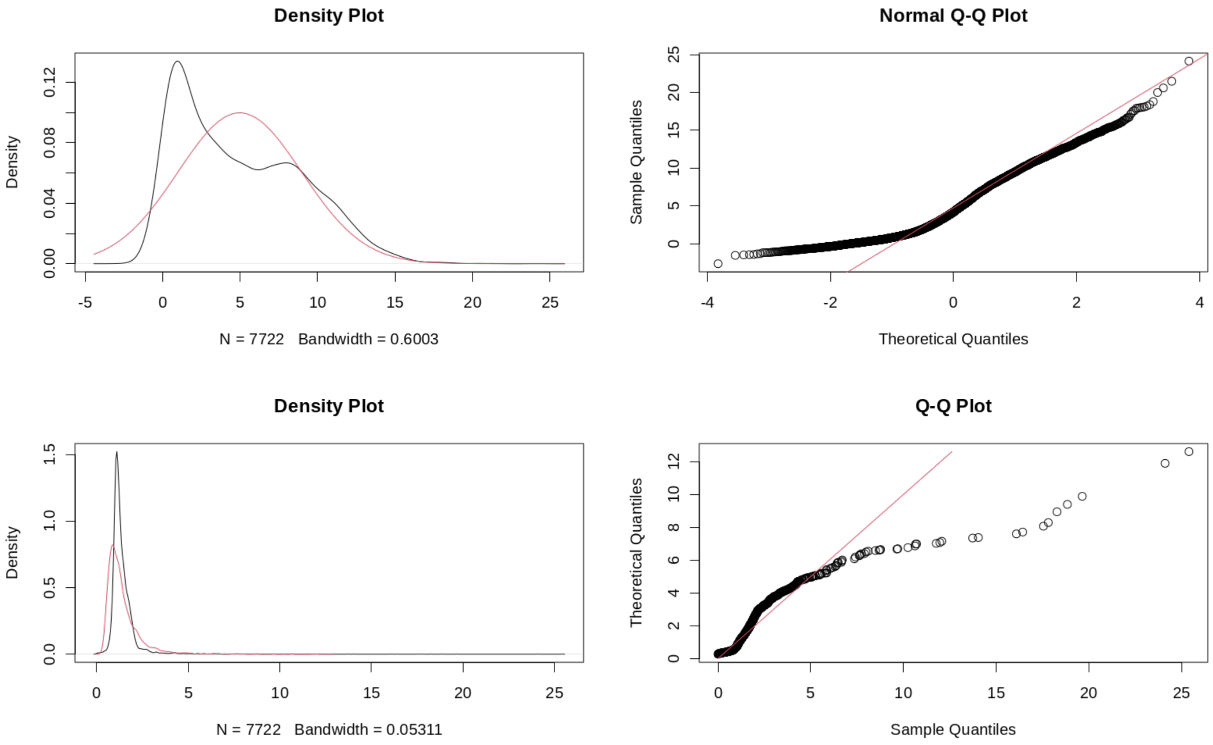

Figure 1: Prior Plots File