MBatch 05-01
Using MBatch Assessments: EBNPlus_CombineBatches
Tod Casasent
2017-11-10-1450

# 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.docx for instructions on downloading test data.

# Algorithm

EBNPlus_CombineBatches is a function used to combine batch information after the data has been combined via the EBNPlus algorithm.

# Output

The primary output method for MBatch is to view results in the Batch Effects Website. Correction algorithms and the combine batches function does not create graphical output and instead creates TSV output files.

# Usage

EBNPlus_CombineBatches(theBeaBatches1, theBeaBatches2, theEBNP_Data1BatchId,

theEBNP_Data2BatchId, theBarcodeTrimFunction = NULL, theSep=".")

# Arguments

**theBeaBatches1** The data.frame containing batch information for data set 1. The "Sample" column should contain barcodes (or other sample ids) and is required.

**theBeaBatches2** The data.frame containing batch information for data set 2. The "Sample" column should contain barcodes (or other sample ids) and is required.

**theEBNP_Data1BatchId** The Batch Id for data set 1, as passed to one of the other EBNPlus functions (for example, RNASeqV2).

**theEBNP_Data2BatchId** The Batch Id for data set 2, as passed to one of the other EBNPlus functions (for example, Agilent4502).

**theBarcodeTrimFunction** A function applied to trim barcodes if needed. This defaults to NULL (indicating no trimming) and should not be needed for TCGA data.

**theSep** Separator used when adding ids to existing barcodes. This defaults to ".". Ids are added to existing sample ids to distinguish between replicates.

# Example Call

The following code combined batch files and is taken from the tests/EBNPlus_CombineBatches.R file. Data used is from the testing data as per the MBatch_01_InstallLinux.docx document.

```
library(MBatch)

# set the paths

theBatchFile="/bea_testing/MATRIX_DATA/brca_rnaseq2_batches.tsv"

theBatchFile2="/bea_testing/MATRIX_DATA/brca_agi4502_batches.tsv"

theOutputDir="/bea_testing/output/EBNPlus_CombineBatches"

theBatchId1="RNASeqV2"

theBatchId2="Agilent4502"

# make sure the output dir exists and is empty

unlink(theOutputDir, recursive=TRUE)

dir.create(theOutputDir, showWarnings=FALSE, recursive=TRUE)

dataBatches <- EBNPlus_CombineBatches(readAsDataFrame(theBatchFile),

readAsDataFrame(theBatchFile2), theBatchId1, theBatchId2)

writeAsDataframe(file.path(theOutputDir, "BatchData.tsv"), dataBatches)
```

## Command Line Output

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.

```
> library(MBatch)

>

> # set the paths

> theBatchFile="/bea_testing/MATRIX_DATA/brca_rnaseq2_batches.tsv"
> theBatchFile2="/bea_testing/MATRIX_DATA/brca_agi4502_batches.tsv"

> theOutputDir="/bea_testing/output/EBNPlus_CombineBatches"

> theBatchId1="RNASeqV2"

> theBatchId2="Agilent4502"

>

> # make sure the output dir exists and is empty

> unlink(theOutputDir, recursive=TRUE)

> dir.create(theOutputDir, showWarnings=FALSE, recursive=TRUE)

>

> dataBatches <- EBNPlus_CombineBatches(readAsDataFrame(theBatchFile),

+ readAsDataFrame(theBatchFile2), theBatchId1, theBatchId2)

2017 10 17 13:42:00.570 DEBUG MachineName starting BeaEBNPlusBatches

2017 10 17 13:42:00.572 DEBUG MachineName readAsDataFrame - thePar
-Xmx2000m

2017 10 17 13:42:00.572 DEBUG MachineName readAsDataFrame - theFile
/bea_testing/MATRIX_DATA/brca_rnaseq2_batches.tsv

2017 10 17 13:42:00.572 DEBUG MachineName readAsDataFrame - Calling .jinit
/home/linux/R/x86_64-pc-linux-gnu-library/3.4/MBatch/ReadRJava/ReadRJava.jar

2017 10 17 13:42:00.579 DEBUG MachineName readAsDataFrame - .jinit complete

2017 10 17 13:42:00.580 DEBUG MachineName readAsDataFrame before java

ReadRJavaL::loadStringData 2014-04-20-1523

2017 10 17 13:42:00.682 DEBUG MachineName readAsDataFrame after java

2017 10 17 13:42:00.685 DEBUG MachineName readAsDataFrame -
length(myData) 7290

ReadRJavaL::loadStringData done

2017 10 17 13:42:00.686 DEBUG MachineName readAsDataFrame -
length(myCols) 6

2017 10 17 13:42:00.686 DEBUG MachineName readAsDataFrame -
length(myRows) 0

2017 10 17 13:42:00.687 DEBUG MachineName readAsDataFrame - myCols
Sample, Type, BatchId, PlateId, ShipDate, TSS

2017 10 17 13:42:00.688 DEBUG MachineName readAsDataFrame - myRows
```

```
2017 10 17 13:42:00.690 DEBUG MachineName readAsDataFrame - thePar -Xmx2000m

2017 10 17 13:42:00.691 DEBUG MachineName readAsDataFrame - theFile /bea_testing/MATRIX_DATA/brca_agi4502_batches.tsv

2017 10 17 13:42:00.691 DEBUG MachineName readAsDataFrame - Calling .jinit /home/linux/R/x86_64-pc-linux-gnu-library/3.4/MBatch/ReadRJava/ReadRJava.jar

2017 10 17 13:42:00.698 DEBUG MachineName readAsDataFrame - .jinit complete

2017 10 17 13:42:00.699 DEBUG MachineName readAsDataFrame before java

ReadRJavaL::loadStringData 2014-04-20-1523

ReadRJavaL::loadStringData done

2017 10 17 13:42:00.703 DEBUG MachineName readAsDataFrame after java

2017 10 17 13:42:00.705 DEBUG MachineName readAsDataFrame - length(myData) 3600

2017 10 17 13:42:00.706 DEBUG MachineName readAsDataFrame - length(myCols) 6

2017 10 17 13:42:00.706 DEBUG MachineName readAsDataFrame - length(myRows) 0

2017 10 17 13:42:00.707 DEBUG MachineName readAsDataFrame - myCols Sample, Type, BatchId, PlateId, ShipDate, TSS

2017 10 17 13:42:00.708 DEBUG MachineName readAsDataFrame - myRows

> writeAsDataframe(file.path(theOutputDir, "BatchData.tsv"), dataBatches)

2017 10 17 13:42:00.747 DEBUG MachineName writeAsDataframe - thePar -Xmx2000m

2017 10 17 13:42:00.748 DEBUG MachineName writeAsDataframe - theFile /bea_testing/output/EBNPlus_CombineBatches/BatchData.tsv

2017 10 17 13:42:00.749 DEBUG MachineName writeAsDataframe - length(myData) 12705

2017 10 17 13:42:00.749 DEBUG MachineName writeAsDataframe - length(myCols) 7

2017 10 17 13:42:00.750 DEBUG MachineName writeAsDataframe - length(myRows) 0

2017 10 17 13:42:00.750 DEBUG MachineName writeAsDataframe - Calling .jinit /home/linux/R/x86_64-pc-linux-gnu-library/3.4/MBatch/ReadRJava/ReadRJava.jar

2017 10 17 13:42:00.756 DEBUG MachineName writeAsDataframe - .jinit complete
```

```
2017 10 17 13:42:00.757 DEBUG MachineName writeAsDataframe before java
ReadRJava::writeStringData_Column 2014-04-20-1523
writeFile - start
writeFile - done
ReadRJava::writeStringData_Column done
2017 10 17 13:42:00.767 DEBUG MachineName writeAsDataframe after java
2017 10 17 13:42:00.767 DEBUG MachineName writeAsDataframe success= TRUE
[1] TRUE
```

## Example File Output

The above code creates the following output files. Files are named using the following naming convention:

BatchData.tsv

This is a TSV file with both original batch files combined and save here.