# *blimaTestingData* - Making data ready to use in the *blima* package *

***Vojtěch Kulvait***[1]

[1] vojtech.kulvait@lf1.cuni.cz

*Thanks to Pavla Jumrová for language corrections.

**November 5, 2019**

## 1 Introduction

*blimaTestingData* has been created mainly to show basic work with a *blima* package. This manual will provide you with a step by step process how to prepare your data for processing with a *blima* package. In fact the core data object is a *list* of *beadLevelData* objects from the package *beadarray*.

The package *blima* was developed to make analysis of Illumina bead level data possible without the need of summarizing them according to probes prior to differential expression testing.

The data in *blimaTestingData* are derived from the NCBI Gene Expression Omnibus dataset GSE56129. The set were prepared using microarray images of human mesenchymal cells treated with various supplements. Set was uploaded primarily to be a supporting dataset for the package blima. The dataset contains 9 Illumina array spots representing 3 different treatment conditions of human mesenchymal cells. First two conditions in quadruplicate and third condition on single array.

```
> library(blimaTestingData)
> data(blimatesting)
```

These are the basic commands to load `blimatesting` object. As was already mentioned this object is simply *list* of two *beadLevelData* objects. Package *blima* and its functions can work with a lists of `beadLevelData` or single `beadLevelData` object as input. This is due to the fact, that analysis such as quantile normalization can be carried out across multiple arrays or array kits. Object `beadLevelData` usually represent one bead array kit with some arrays on it. If user wants to work with multiple kits, they could simply put all the array kits to one list and work with this object inside package *blima*. It is also possible to provide list of logical vectors corresponding to particular arrays to process only selected arrays in the analysis object.

## 2 Preparing `blimatesting` object

This chapter is an step by step description how to prepare object to be used inside *blima* package.

In the https://bitbucket.org/kulvait/blima/downloads/BLIMATESTINGRAW.zip were prepared raw data in usual structure. Let expect we have unzipped BLIMATESTINGRAW.zip in the directory tmpDir = "/tmp/BLIMATESTINGRAW".

In the tmpDir there are two subdirectories 6898481097 and 6898481102 with the scanner text and image files. Each directory contains data from one array kit. It is mandatory to force scanner output to contain txt and tif files to be able to do image processing.

In the tmpDir there are also annotation data for our experiments. There are simply data in tsv (tab separated values) format. These data will work as annotation of our experiments. It is a good practice to prepare such data for each experiment. These files are annotation_6898481097.csv annotation_6898481102.csv. There are also annotations for the chips used in chips.txt. These annotation data does not have to follow exactly the form in the described files these are just examples.

The last directory in tmpDir structure called Illumina contains chip annotation information.

Now we can prepare list of the two objects prepared with the *beadarray* ready to process with the *blima*. Basicly we add annotations of the data to the data objects and process images using readIllumina function. First we load *beadarray* and prepare two helper functions.

```
> library(beadarray)
> processAllData <- function(dir, useImages=FALSE, illuminaAnnotation=NULL,
+                 phenoData, platformName, platformAnnotationPath, processOnly = NULL)
+ {
+         if(missing(dir) | missing(useImages) | missing(illuminaAnnotation)
+                         | missing(phenoData) | missing(platformName)
+                         | missing(platformAnnotationPath))
+         {
+                 stop("Provide all data!")
+         }
+         x <- readIllumina(dir, useImages, illuminaAnnotation, sectionNames=processOnly);
+         x@experimentData$phenoData <- phenoData;
+         x <- insertSectionData(x, what="Platform", data=rep(platformName,
+                                         nrow(x@sectionData$numBeads)))
+         x <- addChipsAnnotation(x, platformAnnotationPath);
+         return(x);
+ }
> addChipsAnnotation <- function(x, platformAnnotationPath)
+ {
+         chipsannotation <- read.AnnotatedDataFrame(platformAnnotationPath);
+         x <- insertSectionData(x, what="Chips", data=pData(chipsannotation))
+         return(x);
+ }
```

Now we create `blimatesting` object and save it as blimatesting.rda file.

```
> inputDirectory = "/tmp/BLIMATESTINGRAW"
> outputDirectory = "/tmp"
> if(file.exists(inputDirectory))
+ {
+     useImages = TRUE;
+     phenox <- read.AnnotatedDataFrame(file.path(inputDirectory, "annotation_6898481097.csv"));
+     phenoy <- read.AnnotatedDataFrame(file.path(inputDirectory, "annotation_6898481102.csv"));
+     px <- pData(phenox)
+     py <- pData(phenoy)
+     px <- px[px$Name %in% c("A1", "A2", "A3", "A4"),]
+     py <- py[py$Name %in% c("D4", "E1", "E2", "E3", "E4"),]
```

```
+     phenox <- AnnotatedDataFrame(data=px)
+     phenoy <- AnnotatedDataFrame(data=py)
+     spotsToReadx <- rownames(px)
+     spotsToReady <- rownames(py)
+     x15 <- processAllData(file.path(inputDirectory,"6898481097"), useImages, "Humanv4", phenox,
+                     "HumanHT-12.v4", file.path(inputDirectory, "chips.txt"), processOnly=spotsToReadx)
+     y15 <- processAllData(file.path(inputDirectory, "6898481102"), useImages, "Humanv4", phenoy,
+                     "HumanHT-12.v4", file.path(inputDirectory, "chips.txt"), processOnly=spotsToReady)
+     blimatesting <- c(x15, y15)
+     save(blimatesting, file=file.path(outputDirectory, "blimatesting.rda"), compress="xz")
+ }
```

# 3 Creating annotation object

Now we create annotation object as a representation of manufacturer text file provided as annotation. For the annotation of the data provided in the *blimaTestingData* package is in Bioconductor the package *illuminaHumanv4.db* but it slightly deviate from the data provided by Illumina in file Illumina/annotation/HumanHT-12_V4_0_R2_15002873_B.txt. So if we have to use original annotation (for example for GEO submission purposes) we should use data from manufacturer annotation file. In the following text we show how to create representation of the manufacturer annotation file as the object `annotationHumanHT12V4`.

```
> if(file.exists(inputDirectory))
+ {
+     annotationHumanHT12V4file <-file.path(inputDirectory,
+                     "Illumina/annotation/HumanHT-12_V4_0_R2_15002873_B.txt")
+     annotationLines <- readLines(annotationHumanHT12V4file)
+     sections <- grep("^\\[.*\\]$",annotationLines)
+     sections <- c(sections, length(annotationLines) + 1)
+     annotationHumanHT12V4 <- list()
+     for(i in 1:(length(sections)-1))
+     {
+             row <- sections[i]
+             sectionName <- substr(annotationLines[row], 2, nchar(annotationLines[row])-1)
+             if(sectionName == "Heading")
+             {
+                     section <- read.csv(annotationHumanHT12V4file, sep="\t", skip=row,
+                                     nrows = sections[i+1] - sections[i] - 1, header=FALSE)
+             }else
+             {
+                     section <- read.csv(annotationHumanHT12V4file, sep="\t", skip=row,
+                                     nrows = sections[i+1] - sections[i] - 1)
+             }
+
+             annotationHumanHT12V4[[length(annotationHumanHT12V4) + 1]] = section
+             names(annotationHumanHT12V4)[i] <- sectionName
+     }
+     save(annotationHumanHT12V4, file=file.path(outputDirectory,
+                     "annotationHumanHT12V4.rda"), compress="xz")
```

```
+ }
```

This is it! We can now load objects we have created by calling

```
> if(file.exists(inputDirectory))
+ {
+     load(file=file.path(outputDirectory, "blimatesting.rda"), envir=.GlobalEnv)
+     load(file=file.path(outputDirectory, "annotationHumanHT12V4.rda"), envir=.GlobalEnv)
+ }
```

Object `blimatesting` is also part of the package *blimaTestingData* and you can load it by

```
> data(blimatesting)
```

Object `annotationHumanHT12V4` was excluded from the package not to confuse data and annotation packages. You can create it by following this guide or you can use some existing annotation package.