MetaGxBreast: a package for breast cancer gene expression analysis

Michael Zon[1], Deena M.A. Gendoo[1,2], Natchar Ratanasirigulchai[1], Gregory Chen[2], Levi Waldron[3,4], and Benjamin Haibe-Kains[*1,2]

[1]Bioinformatics and Computational Genomics Laboratory, Princess Margaret Cancer Center, University Health Network, Toronto, Ontario, Canada
[2]Department of Medical Biophysics, University of Toronto, Toronto, Canada
[3]Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute, Boston, MA, USA
[4]Department of Biostatistics, Harvard School of Public Health, Boston, MA, USA

February 4, 2021

Contents

[*]benjamin.haibe.kains@utoronto.ca

1 Installing the Package

The MetaGxBreast package is a compendium of Breast Cancer datasets. The package is publicly available and can be installed from Bioconductor into R version 3.5.0 or higher.

To install the MetaGxBreast package from Bioconductor:

```
> if (!requireNamespace("BiocManager", quietly = TRUE))
+     install.packages("BiocManager")
> BiocManager::install("MetaGxBreast")
```

2 Loading Datasets

First we load the MetaGxBreast package into the workspace.

To load the packages into R and obtain some datasets, please use the following commands:

```
> library(MetaGxBreast)
> esets = MetaGxBreast::loadBreastEsets(loadString = c("CAL", "DFHCC", "DFHCC2", "DFH
```

This will load 7 of the 37 expression datasets. Users can modify the parameters of the function to restrict datasets that do not meet certain criteria for loading. Also note that loadString = "majority" will load 37 of the 39 datasets. The larger metabric and tcga studies need to be loaded separately by altering the loadString variable to include the string metabric or tcga. Some example parameters are shown below:

Datasets: Retain only genes that are common across all platforms loaded (default = FALSE)

Datasets: Retain studies with a minimum sample size (default = 0)

Datasets: Retain studies with a minimum umber of genes (default = 0)

Datasets: Retain studies with a minimum number of survival events (default = 0)

Datasets: Remove duplicate samples (default = TRUE)

3 Obtaining Sample Counts in Datasets

To obtain the number of samples per dataset, run the following:

```
> numSamples <- vapply(seq_along(esets), FUN=function(i, esets){
+   length(sampleNames(esets[[i]]))
+   }, numeric(1), esets=esets)
> SampleNumberSummaryAll <- data.frame(NumberOfSamples = numSamples,
+                                      row.names = names(esets))
> total <- sum(SampleNumberSummaryAll[,"NumberOfSamples"])
> SampleNumberSummaryAll <- rbind(SampleNumberSummaryAll, total)
> rownames(SampleNumberSummaryAll)[nrow(SampleNumberSummaryAll)] <- "Total"
> require(xtable)
> print(xtable(SampleNumberSummaryAll,digits = 2), floating = FALSE)
```

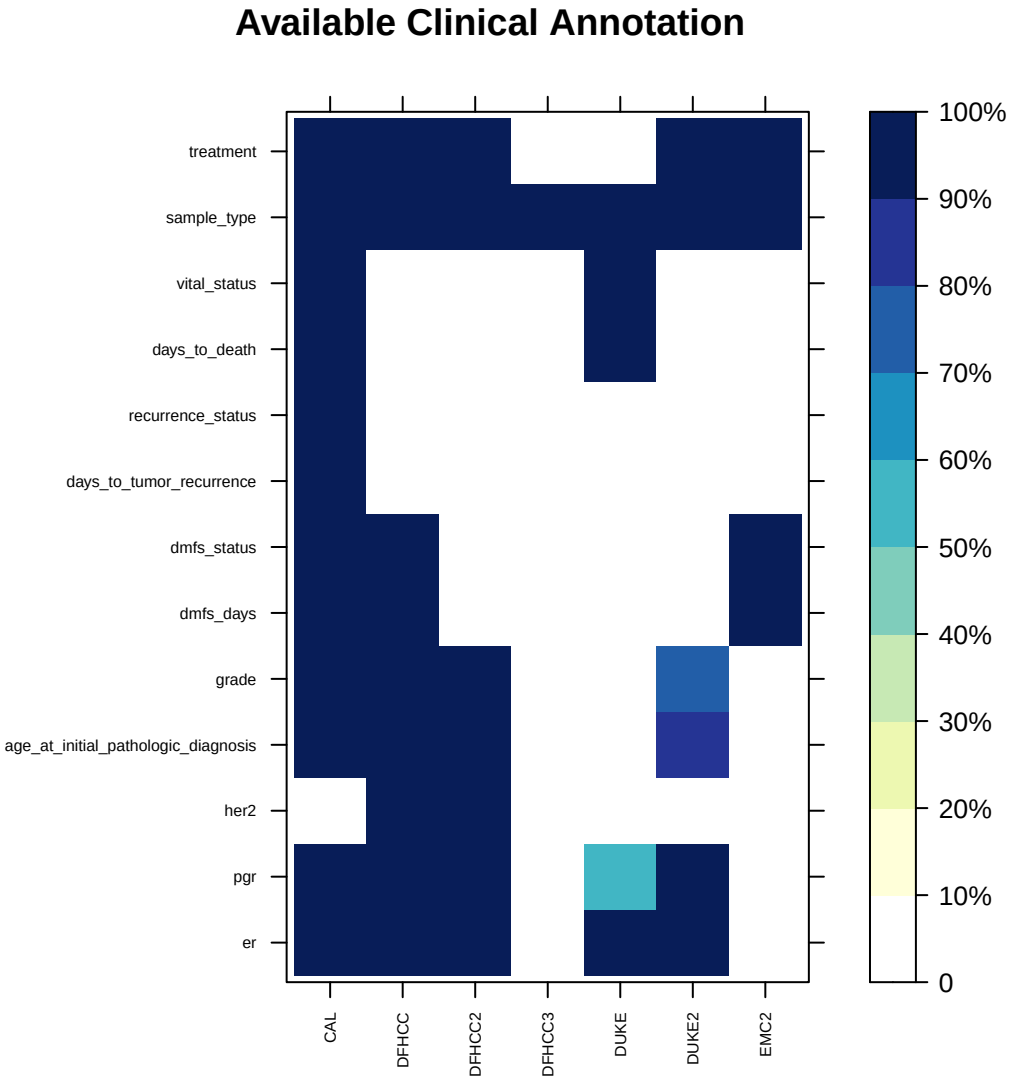
	NumberOfSamples
CAL	118.00
DFHCC	115.00
DFHCC2	83.00
DFHCC3	40.00
DUKE	169.00
DUKE2	154.00
EMC2	204.00
Total	883.00

4 Assess Phenotype Data

We can also obtain a summary of the phenotype data (pData) for each expression dataset. Here, we assess the proportion of samples in every datasets that contain a specific pData variable.

```
> #pData Variables
> pDataID <- c("er","pgr", "her2", "age_at_initial_pathologic_diagnosis",
+              "grade", "dmfs_days", "dmfs_status", "days_to_tumor_recurrence",
+              "recurrence_status", "days_to_death", "vital_status", "sample_type", "
> pDataPercentSummaryTable <- NULL
> pDataSummaryNumbersTable <- NULL
> pDataSummaryNumbersList = lapply(esets, function(x)
+   vapply(pDataID, function(y) sum(!is.na(pData(x)[,y])), numeric(1)))
> pDataPercentSummaryList = lapply(esets, function(x)
+   vapply(pDataID, function(y)
```

```
+     sum(!is.na(pData(x)[,y]))/nrow(pData(x)), numeric(1))*100)
> pDataSummaryNumbersTable = sapply(pDataSummaryNumbersList, function(x) x)
> pDataPercentSummaryTable = sapply(pDataPercentSummaryList, function(x) x)
> rownames(pDataSummaryNumbersTable) <- pDataID
> rownames(pDataPercentSummaryTable) <- pDataID
> colnames(pDataSummaryNumbersTable) <- names(esets)
> colnames(pDataPercentSummaryTable) <- names(esets)
> pDataSummaryNumbersTable <- rbind(pDataSummaryNumbersTable, total)
> rownames(pDataSummaryNumbersTable)[nrow(pDataSummaryNumbersTable)] <- "Total"
> # Generate a heatmap representation of the pData
> pDataPercentSummaryTable<-t(pDataPercentSummaryTable)
> pDataPercentSummaryTable<-cbind(Name=(rownames(pDataPercentSummaryTable))
+                                 ,pDataPercentSummaryTable)
> nba<-pDataPercentSummaryTable
> gradient_colors = c("#ffffff","#ffffd9","#edf8b1","#c7e9b4","#7fcdbb",
+                     "#41b6c4","#1d91c0","#225ea8","#253494","#081d58")
> library(lattice)
> nbamat<-as.matrix(nba)
> rownames(nbamat)<-nbamat[,1]
> nbamat<-nbamat[,-1]
> Interval<-as.numeric(c(10,20,30,40,50,60,70,80,90,100))
> levelplot(nbamat,col.regions=gradient_colors,
+           main="Available Clinical Annotation",
+           scales=list(x=list(rot=90, cex=0.5),
+                       y= list(cex=0.5),key=list(cex=0.2)),
+           at=seq(from=0,to=100,length=10),
+           cex=0.2, ylab="", xlab="", lattice.options=list(),
+           colorkey=list(at=as.numeric(factor(c(seq(from=0, to=100, by=10)))),
+                   labels=as.character(c( "0","10%","20%","30%", "40%","50%",
+                                          "60%", "70%", "80%","90%", "100%"),
+                                       cex=0.2,font=1,col="brown",height=1,
+                                       width=1.4), col=(gradient_colors)))
>
```



5 Session Info

- R Under development (unstable) (2021-01-19 r79848), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`,

LC_MESSAGES=en_US.UTF-8, LC_PAPER=en_US.UTF-8, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=en_US.UTF-8, LC_IDENTIFICATION=C

- Running under: Ubuntu 20.04.2 LTS
- Matrix products: default
- BLAS: /home/biocbuild/bbs-3.13-bioc/R/lib/libRblas.so
- LAPACK: /home/biocbuild/bbs-3.13-bioc/R/lib/libRlapack.so
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: AnnotationHub 2.23.1, Biobase 2.51.0, BiocFileCache 1.15.1, BiocGenerics 0.37.1, ExperimentHub 1.17.0, GenomeInfoDb 1.27.5, GenomicRanges 1.43.3, IRanges 2.25.6, MatrixGenerics 1.3.1, MetaGxBreast 1.11.0, S4Vectors 0.29.7, SummarizedExperiment 1.21.1, dbplyr 2.0.0, impute 1.65.0, lattice 0.20-41, matrixStats 0.58.0, xtable 1.8-4
- Loaded via a namespace (and not attached): AnnotationDbi 1.53.0, BiocManager 1.30.10, BiocVersion 3.13.1, DBI 1.1.1, DelayedArray 0.17.7, GenomeInfoDbData 1.2.4, Matrix 1.3-2, R6 2.5.0, RCurl 1.98-1.2, RSQLite 2.2.3, Rcpp 1.0.6, XVector 0.31.1, assertthat 0.2.1, bit 4.0.4, bit64 4.0.5, bitops 1.0-6, blob 1.2.1, cachem 1.0.2, compiler 4.1.0, crayon 1.4.0, curl 4.3, digest 0.6.27, dplyr 1.0.4, ellipsis 0.3.1, fastmap 1.1.0, filelock 1.0.2, generics 0.1.0, glue 1.4.2, grid 4.1.0, htmltools 0.5.1.1, httpuv 1.5.5, httr 1.4.2, interactiveDisplayBase 1.29.0, later 1.1.0.1, lifecycle 0.2.0, magrittr 2.0.1, memoise 2.0.0, mime 0.9, pillar 1.4.7, pkgconfig 2.0.3, promises 1.1.1, purrr 0.3.4, rappdirs 0.3.3, rlang 0.4.10, shiny 1.6.0, tibble 3.0.6, tidyselect 1.1.0, tools 4.1.0, vctrs 0.3.6, withr 2.4.1, yaml 2.2.1, zlibbioc 1.37.0