

# MIGSA: Getting pbcmc datasets

**Juan C Rodriguez**

CONICET  
Universidad Católica de Córdoba  
Universidad Nacional de Córdoba

**Cristóbal Fresno**

Instituto Nacional de Medicina Genómica

**Andrea S Llera**

CONICET  
Fundación Instituto Leloir

**Elmer A Fernández**

CONICET  
Universidad Católica de Córdoba  
Universidad Nacional de Córdoba

---

## Abstract

In this vignette we are going to show how we got the RData *pbcmcData.RData* which can be loaded via the **MIGSAdata** package using `data(pbcmcData)`.

*Keywords:* singular enrichment analysis, over representation analysis, gene set enrichment analysis, functional class scoring, big omics data.

---

## 1. Getting the data

Following we give the used code to download this data and their PAM50 subtypes.

```
> library(limma);
> library(pbcmc);
> # datasets included in BioConductor repository
> libNames <- c("mainz", "nki", "transbig", "unt", "upp", "vdx");
> # let's load them!
> pbcmcData <- lapply(libNames, function(actLibName) {
+   print(actLibName);
+
+   # the pbcmc package provides an easy way to download and classify them
+   actLib <- loadBCDataset(Class=PAM50, libname=actLibName, verbose=FALSE);
+   actLibFilt <- filtrate(actLib, verbose=FALSE);
+   actLibFilt <- classify(actLibFilt, std="none", verbose=FALSE);
+   actSubtypes <- classification(actLibFilt)$subtype;
+
+   # get the expression matrix and the annotation
+   actExprs <- exprs(actLib);
+   actAnnot <- annotation(actLib);
+ }
```

```

+   # we recommend working allways with Entrez IDs, let's match them with
+   # expression matrix rownames (and modify them)
+   if (all(actAnnot$probe == rownames(actExprs))) {
+     actExprs <- actExprs[!is.na(actAnnot$EntrezGene.ID),];
+     actAnnot <- actAnnot[!is.na(actAnnot$EntrezGene.ID),];
+     rownames(actExprs) <- as.character(actAnnot$EntrezGene.ID);
+   } else {
+     matchedEntrez <- match(rownames(actExprs), actAnnot$probe);
+     # all(rownames(actExprs) %in% actAnnot$probe == !is.na(matchedEntrez));
+
+     stopifnot(all(
+       actAnnot$probe[!is.na(matchedEntrez)] ==
+       rownames(actExprs)[!is.na(matchedEntrez)]));
+
+     actExprs <- actExprs[!is.na(matchedEntrez),];
+     actAnnot <- actAnnot[!is.na(matchedEntrez),];
+     stopifnot(all(actAnnot$probe == rownames(actExprs)));
+     actExprs <- actExprs[!is.na(actAnnot$EntrezGene.ID),];
+     actAnnot <- actAnnot[!is.na(actAnnot$EntrezGene.ID),];
+     rownames(actExprs) <- as.character(actAnnot$EntrezGene.ID);
+   }
+
+   # average repeated genes expression
+   actExprs <- avereps(actExprs);
+
+   stopifnot(all(colnames(actExprs) == names(actSubtypes)));
+   # filtrate only these two conditions
+   actExprs <- actExprs[, actSubtypes %in% c("Basal", "LumA")];
+   actSubtypes <- as.character(
+     actSubtypes[actSubtypes %in% c("Basal", "LumA")]);
+
+   return(list(geneExpr=actExprs, subtypes=actSubtypes));
+ })
> names(pbcmcData) <- libNames;

```

And let's check it is the same data.

```

> # save the just created pbcmcData to newPbcmcData
> newPbcmcData <- pbcmcData;
> library(MIGSAdata);
> # and load the MIGSAdata one.
> data(pbcmcData);
> all.equal(newPbcmcData, pbcmcData);

```

## Session Info

```
> sessionInfo()
```

```
R version 4.0.0 (2020-04-24)
```

```
Platform: x86_64-pc-linux-gnu (64-bit)
```

```
Running under: Ubuntu 18.04.4 LTS
```

```
Matrix products: default
```

```
BLAS: /home/biocbuild/bbs-3.11-bioc/R/lib/libRblas.so
```

```
LAPACK: /home/biocbuild/bbs-3.11-bioc/R/lib/libRlapack.so
```

```
locale:
```

```
[1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
[3] LC_TIME=en_US.UTF-8      LC_COLLATE=C
[5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8     LC_NAME=C
[9] LC_ADDRESS=C             LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
```

```
attached base packages:
```

```
[1] stats4      parallel  stats      graphics  grDevices  utils      datasets
[8] methods    base
```

```
other attached packages:
```

```
[1] edgeR_3.30.0      MIGSAdata_1.11.0    MIGSA_1.12.0
[4] mGSZ_1.0          ismev_1.42          mgcv_1.8-31
[7] nlme_3.1-147      MASS_7.3-51.6       limma_3.44.0
[10] GSA_1.03.1        BiocParallel_1.22.0 GSEABase_1.50.0
[13] graph_1.66.0      annotate_1.66.0      XML_3.99-0.3
[16] AnnotationDbi_1.50.0 IRanges_2.22.0      S4Vectors_0.26.0
[19] Biobase_2.48.0    BiocGenerics_0.34.0
```

```
loaded via a namespace (and not attached):
```

```
[1] gg dendro_0.1-20      bit64_0.9-7          splines_4.0.0
[4] assertthat_0.2.1     RBGL_1.64.0          blob_1.2.1
[7] Category_2.54.0      pillar_1.4.3         RSQLite_2.2.0
[10] lattice_0.20-41      glue_1.4.0           digest_0.6.25
[13] colorspace_1.4-1     Matrix_1.2-18        plyr_1.8.6
[16] pkgconfig_2.0.3      genefilter_1.70.0    purrr_0.3.4
[19] xtable_1.8-4         GO.db_3.10.0         scales_1.1.0
[22] tibble_3.0.1         farver_2.0.3         ggplot2_3.3.0
[25] ellipsis_0.3.0      survival_3.1-12      RJSONIO_1.3-1.4
[28] magrittr_1.5         crayon_1.3.4         memoise_1.1.0
[31] GOstats_2.54.0       vegan_2.5-6          tools_4.0.0
[34] data.table_1.12.8    org.Hs.eg.db_3.10.0  formatR_1.7
[37] lifecycle_0.2.0     matrixStats_0.56.0   stringr_1.4.0
[40] munsell_0.5.0        locfit_1.5-9.4       cluster_2.1.0
[43] lambda.r_1.2.4       compiler_4.0.0       rlang_0.4.5
```

```
[46] futile.logger_1.4.3    grid_4.0.0          RCurl_1.98-1.2
[49] AnnotationForge_1.30.0  labeling_0.3        bitops_1.0-6
[52] gtable_0.3.0           DBI_1.1.0           reshape2_1.4.4
[55] R6_2.4.1               dplyr_0.8.5         bit_1.1-15.2
[58] futile.options_1.0.1    permute_0.9-5       Rgraphviz_2.32.0
[61] stringi_1.4.6          Rcpp_1.0.4.6        vctrs_0.2.4
[64] tidycselect_1.0.0
```

**Affiliation:**

Juan C Rodriguez & Elmer A Fernández

Bioscience Data Mining Group

Facultad de Ingeniería

Universidad Católica de Córdoba - CONICET

X5016DHK Córdoba, Argentina

E-mail: [jcrodriguez@bdmg.com.ar](mailto:jcrodriguez@bdmg.com.ar), [efernandez@bdmg.com.ar](mailto:efernandez@bdmg.com.ar)

URL: <http://www.bdmg.com.ar/>