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 3.6.2 (2019-12-12)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 18.04.3 LTS

Matrix products: default
BLAS:   /home/biocbuild/bbs-3.10-bioc/R/lib/libRblas.so
LAPACK: /home/biocbuild/bbs-3.10-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_3.28.0         MIGSAdata_1.10.0     MIGSA_1.10.1
 [4] mGSZ_1.0             ismev_1.42           mgcv_1.8-31
 [7] nlme_3.1-143         MASS_7.3-51.5        limma_3.42.0
[10] GSA_1.03.1           BiocParallel_1.20.1  GSEABase_1.48.0
[13] graph_1.64.0         annotate_1.64.0      XML_3.98-1.20
[16] AnnotationDbi_1.48.0 IRanges_2.20.1       S4Vectors_0.24.1
[19] Biobase_2.46.0       BiocGenerics_0.32.0

loaded via a namespace (and not attached):
 [1] ggdendro_0.1-20        bit64_0.9-7            splines_3.6.2
 [4] assertthat_0.2.1       RBGL_1.62.1            blob_1.2.0
 [7] Category_2.52.1        pillar_1.4.3           RSQLite_2.2.0
[10] backports_1.1.5        lattice_0.20-38        glue_1.3.1
[13] digest_0.6.23          colorspace_1.4-1       Matrix_1.2-18
[16] plyr_1.8.5             pkgconfig_2.0.3        genefilter_1.68.0
[19] purrr_0.3.3            xtable_1.8-4           GO.db_3.10.0
[22] scales_1.1.0           tibble_2.1.3           farver_2.0.1
[25] ggplot2_3.2.1          lazyeval_0.2.2         survival_3.1-8
[28] RJSONIO_1.3-1.3        magrittr_1.5           crayon_1.3.4
[31] memoise_1.1.0          GOstats_2.52.0         vegan_2.5-6
[34] tools_3.6.2            data.table_1.12.8      org.Hs.eg.db_3.10.0
[37] formatR_1.7            lifecycle_0.1.0        matrixStats_0.55.0
[40] stringr_1.4.0          munsell_0.5.0          locfit_1.5-9.1
[43] cluster_2.1.0          lambda.r_1.2.4         compiler_3.6.2
```

```
[46] rlang_0.4.2             futile.logger_1.4.3    grid_3.6.2
[49] RCurl_1.95-4.12         AnnotationForge_1.28.0 labeling_0.3
[52] bitops_1.0-6            gtable_0.3.0           DBI_1.1.0
[55] reshape2_1.4.3          R6_2.4.1               dplyr_0.8.3
[58] bit_1.1-14              zeallot_0.1.0          futile.options_1.0.1
[61] permute_0.9-5           Rgraphviz_2.30.0       stringi_1.4.3
[64] Rcpp_1.0.3              vctrs_0.2.1            tidyselect_0.2.5
```

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, efernandez@bdmg.com.ar
URL: http://www.bdmg.com.ar/