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 3.6.0 (2019-04-26)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows Server 2012 R2 x64 (build 9600)

Matrix products: default

locale:
[1] LC_COLLATE=C
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] stats4    parallel  stats     graphics  grDevices utils     datasets
[8] methods   base

other attached packages:
 [1] edgeR_3.26.5         MIGSAdata_1.8.0      MIGSA_1.8.1
 [4] mGSZ_1.0             ismev_1.42           mgcv_1.8-28
 [7] nlme_3.1-140         MASS_7.3-51.4        limma_3.40.2
[10] GSA_1.03.1           BiocParallel_1.18.0  GSEABase_1.46.0
[13] graph_1.62.0         annotate_1.62.0      XML_3.98-1.20
[16] AnnotationDbi_1.46.0 IRanges_2.18.1       S4Vectors_0.22.0
[19] Biobase_2.44.0       BiocGenerics_0.30.0

loaded via a namespace (and not attached):
 [1] ggdendro_0.1-20        bit64_0.9-7            splines_3.6.0
 [4] assertthat_0.2.1       RBGL_1.60.0            blob_1.2.0
 [7] Category_2.50.0        pillar_1.4.2           RSQLite_2.1.1
[10] backports_1.1.4        lattice_0.20-38        glue_1.3.1
[13] digest_0.6.20          colorspace_1.4-1       Matrix_1.2-17
[16] plyr_1.8.4             pkgconfig_2.0.2        genefilter_1.66.0
[19] purrr_0.3.2            xtable_1.8-4           GO.db_3.8.2
[22] snow_0.4-3             scales_1.0.0           tibble_2.1.3
[25] ggplot2_3.2.0          lazyeval_0.2.2         survival_2.44-1.1
[28] RJSONIO_1.3-1.2        magrittr_1.5           crayon_1.3.4
[31] memoise_1.1.0          GOstats_2.50.0         vegan_2.5-5
[34] tools_3.6.0            data.table_1.12.2      org.Hs.eg.db_3.8.2
[37] formatR_1.7            matrixStats_0.54.0     stringr_1.4.0
[40] munsell_0.5.0          locfit_1.5-9.1         cluster_2.1.0
[43] lambda.r_1.2.3         compiler_3.6.0         rlang_0.4.0
[46] futile.logger_1.4.3    grid_3.6.0             RCurl_1.95-4.12
[49] AnnotationForge_1.26.0 labeling_0.3           bitops_1.0-6
[52] gtable_0.3.0           DBI_1.0.0              reshape2_1.4.3
```

```
[55] R6_2.4.0                dplyr_0.8.3             bit_1.1-14
[58] zeallot_0.1.0           futile.options_1.0.1    permute_0.9-5
[61] Rgraphviz_2.28.0        stringi_1.4.3           Rcpp_1.0.1
[64] vctrs_0.2.0             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/