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 4.0.3 (2020-10-10)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Mojave 10.14.6

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib

locale:
[1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

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

other attached packages:
 [1] edgeR_3.32.0         MIGSAdata_1.14.0     MIGSA_1.14.1
 [4] mGSZ_1.0             ismev_1.42           mgcv_1.8-33
 [7] nlme_3.1-150         MASS_7.3-53          limma_3.46.0
[10] GSA_1.03.1           BiocParallel_1.24.0  GSEABase_1.52.0
[13] graph_1.68.0         annotate_1.68.0      XML_3.99-0.5
[16] AnnotationDbi_1.52.0 IRanges_2.24.0       S4Vectors_0.28.0
[19] Biobase_2.50.0       BiocGenerics_0.36.0

loaded via a namespace (and not attached):
 [1] ggdendro_0.1.22        httr_1.4.2             bit64_4.0.5
 [4] jsonlite_1.7.1         splines_4.0.3          RBGL_1.66.0
 [7] blob_1.2.1             Category_2.56.0        pillar_1.4.6
[10] RSQLite_2.2.1          lattice_0.20-41        glue_1.4.2
[13] digest_0.6.27          colorspace_1.4-1       Matrix_1.2-18
[16] plyr_1.8.6             pkgconfig_2.0.3        genefilter_1.72.0
[19] purrr_0.3.4            xtable_1.8-4           GO.db_3.12.0
[22] scales_1.1.1           tibble_3.0.4           farver_2.0.3
[25] generics_0.0.2         ggplot2_3.3.2          ellipsis_0.3.1
[28] survival_3.2-7         magrittr_1.5           crayon_1.3.4
[31] memoise_1.1.0          GOstats_2.56.0         vegan_2.5-6
[34] tools_4.0.3            data.table_1.13.2      org.Hs.eg.db_3.12.0
[37] formatR_1.7            lifecycle_0.2.0        matrixStats_0.57.0
[40] stringr_1.4.0          munsell_0.5.0          locfit_1.5-9.4
[43] cluster_2.1.0          lambda.r_1.2.4         compiler_4.0.3
[46] rlang_0.4.8            futile.logger_1.4.3    grid_4.0.3
[49] RCurl_1.98-1.2         AnnotationForge_1.32.0 labeling_0.4.2
[52] bitops_1.0-6           gtable_0.3.0           DBI_1.1.0
[55] reshape2_1.4.4         R6_2.5.0               dplyr_1.0.2
[58] bit_4.0.4              futile.options_1.0.1   permute_0.9-5
```

```
[61] Rgraphviz_2.34.0       stringi_1.5.3          Rcpp_1.0.5
[64] vctrs_0.3.4            tidyselect_1.1.0
```

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