MIGSA: Getting TCGA 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 *tcgaMAdata.RData* which can be loaded via the **MIGSAdata** package using data(tcgaMAdata) and *tcgaRNAseqData.RData* which can be loaded using data(tcgaRNAseqData).

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

1. Getting the data

From the TCGA data portal the breast invasive carcinoma (BRCA) microarray and RNAseq datasets present at the date were downloaded. PAM50 subtypes Basal vs. Luminal A were evaluated. With these subjects, *tcgaMAdata.RData* and *tcgaRNAseqData.RData* were built.

1.1. Basal-like subjects

Basal-like TCGA subjects identifiers used:

```
> library(MIGSAdata);
> data(tcgaMAdata);
> names(tcgaMAdata$subtypes)[ tcgaMAdata$subtypes == "Basal" ];

 [1] "TCGA-BH-A0E0-01A-11R-A056-07" "TCGA-A0-A0JL-01A-11R-A056-07"
 [3] "TCGA-B6-A0RE-01A-11R-A056-07" "TCGA-BH-A0B3-01A-11R-A056-07"
 [5] "TCGA-AN-A0AL-01A-11R-A00Z-07" "TCGA-A2-A0YJ-01A-11R-A109-07"
 [7] "TCGA-AR-A0U4-01A-11R-A109-07" "TCGA-B6-A0WX-01A-11R-A109-07"
 [9] "TCGA-A2-A0YM-01A-11R-A109-07" "TCGA-A2-A0YE-01A-11R-A109-07"
[11] "TCGA-B6-A0X1-01A-11R-A109-07" "TCGA-AR-A0TU-01A-31R-A109-07"
[13] "TCGA-BH-A0WA-01A-11R-A109-07" "TCGA-AN-A0XU-01A-11R-A109-07"
```

```
[15] "TCGA-AR-A0U0-01A-11R-A109-07" "TCGA-E2-A108-01A-13R-A10J-07"
[17] "TCGA-AO-A124-01A-11R-A10J-07" "TCGA-AO-A128-01A-11R-A10J-07"
[19] "TCGA-AO-A129-01A-21R-A10J-07" "TCGA-BH-A0BG-01A-11R-A115-07"
[21] "TCGA-BH-A0DL-01A-11R-A115-07" "TCGA-AO-A12F-01A-11R-A115-07"
[23] "TCGA-C8-A131-01A-11R-A115-07" "TCGA-E2-A14R-01A-11R-A115-07"
[25] "TCGA-BH-A0BL-01A-11R-A115-07" "TCGA-AR-A0U1-01A-11R-A115-07"
[27] "TCGA-A2-A04U-01A-11R-A115-07" "TCGA-C8-A12K-01A-21R-A115-07"
[29] "TCGA-C8-A134-01A-11R-A115-07" "TCGA-D8-A142-01A-11R-A115-07"
[31] "TCGA-E2-A14X-01A-11R-A115-07" "TCGA-C8-A12V-01A-11R-A115-07"
[33] "TCGA-D8-A143-01A-11R-A115-07" "TCGA-BH-A0AV-01A-31R-A115-07"
[35] "TCGA-BH-A0BW-01A-11R-A115-07" "TCGA-A7-A0DA-01A-31R-A115-07"
[37] "TCGA-D8-A147-01A-11R-A115-07" "TCGA-BH-A0BW-11A-12R-A115-07"
[39] "TCGA-AR-A0TS-01A-11R-A115-07" "TCGA-D8-A13Z-01A-11R-A115-07"
[41] "TCGA-A8-A08H-01A-21R-A00Z-07" "TCGA-A2-A0D0-01A-11R-A00Z-07"
[43] "TCGA-AN-A0FJ-01A-11R-A00Z-07" "TCGA-A8-A07O-01A-11R-A00Z-07"
[45] "TCGA-AN-A0AT-01A-11R-A034-07" "TCGA-A2-A0CM-01A-31R-A034-07"
[47] "TCGA-A2-A04T-01A-21R-A034-07" "TCGA-A7-A0CE-01A-11R-A00Z-07"
[49] "TCGA-AO-A03R-01A-21R-A034-07" "TCGA-AN-A04D-01A-21R-A034-07"
[51] "TCGA-AQ-A04J-01A-02R-A034-07" "TCGA-A2-A04P-01A-31R-A034-07"
[53] "TCGA-A2-A04Q-01A-21R-A034-07" "TCGA-A8-A07C-01A-11R-A034-07"
[55] "TCGA-A8-A07R-01A-21R-A034-07" "TCGA-A8-A07U-01A-11R-A034-07"
[57] "TCGA-A8-A08R-01A-11R-A034-07" "TCGA-A2-A0D2-01A-21R-A034-07"
[59] "TCGA-BH-A0E6-01A-11R-A034-07" "TCGA-AN-A0FL-01A-11R-A034-07"
[61] "TCGA-AN-A0FX-01A-11R-A034-07" "TCGA-AN-A0G0-01A-11R-A034-07"
[63] "TCGA-B6-A0I2-01A-11R-A034-07" "TCGA-B6-A0I6-01A-11R-A034-07"
[65] "TCGA-B6-A0IJ-01A-11R-A034-07" "TCGA-AO-A0J4-01A-11R-A034-07"
[67] "TCGA-AO-A0J6-01A-11R-A034-07" "TCGA-B6-A0IQ-01A-11R-A034-07"
[69] "TCGA-BH-A0RX-01A-21R-A084-07" "TCGA-A2-A0ST-01A-12R-A084-07"
[71] "TCGA-B6-A0RU-01A-11R-A084-07" "TCGA-A1-A0SO-01A-22R-A084-07"
[73] "TCGA-A2-A0T0-01A-22R-A084-07" "TCGA-A1-A0SP-01A-11R-A084-07"
[75] "TCGA-A1-A0SK-01A-12R-A084-07" "TCGA-A2-A0SX-01A-12R-A084-07"
[77] "TCGA-B6-A0RT-01A-21R-A084-07" "TCGA-AR-A0TP-01A-11R-A084-07"
[79] "TCGA-A7-A0DB-11A-33R-A089-07" "TCGA-A2-A0T2-01A-11R-A084-07"
[81] "TCGA-AR-A1AH-01A-11R-A12D-07" "TCGA-E2-A158-01A-11R-A12D-07"
[83] "TCGA-BH-A18V-01A-11R-A12D-07" "TCGA-E2-A14Y-01A-21R-A12D-07"
[85] "TCGA-E2-A150-01A-11R-A12D-07" "TCGA-BH-A18G-01A-11R-A12D-07"
[87] "TCGA-BH-A18Q-01A-12R-A12D-07" "TCGA-A7-A13D-01A-13R-A12P-07"
[89] "TCGA-AR-A1AO-01A-11R-A12P-07" "TCGA-A7-A13E-01A-11R-A12P-07"
[91] "TCGA-AR-A1AY-01A-21R-A12P-07" "TCGA-AR-A1AQ-01A-11R-A12P-07"
[93] "TCGA-AR-A1AR-01A-31R-A137-07" "TCGA-E2-A14N-01A-31R-A137-07"
[95] "TCGA-BH-A1F0-01A-11R-A137-07"
```

1.2. Luminal A subjects

Luminal A TCGA subjects identifiers used:

```
> library(MIGSAdata);
```

```
> data(tcgaMAdata);
> names(tcgaMAdata$subtypes)[ tcgaMAdata$subtypes == "LumA" ];

  [1] "TCGA-BH-A0BA-01A-11R-A056-07" "TCGA-BH-A0DS-01A-11R-A056-07"
  [3] "TCGA-BH-A0H6-01A-21R-A056-07" "TCGA-AO-A0JJ-01A-11R-A056-07"
  [5] "TCGA-A8-A0A6-01A-12R-A056-07" "TCGA-BH-A0BJ-01A-11R-A056-07"
  [7] "TCGA-BH-A0DP-01A-21R-A056-07" "TCGA-A8-A08O-01A-21R-A056-07"
  [9] "TCGA-A8-A0AD-01A-11R-A056-07" "TCGA-BH-A0BM-01A-11R-A056-07"
 [11] "TCGA-BH-A0HF-01A-11R-A056-07" "TCGA-A7-A0D9-01A-31R-A056-07"
 [13] "TCGA-BH-A0HK-01A-11R-A056-07" "TCGA-BH-A0GZ-01A-11R-A056-07"
 [15] "TCGA-AO-A0JF-01A-11R-A056-07" "TCGA-B6-A0WT-01A-11R-A109-07"
 [17] "TCGA-AN-A0XV-01A-11R-A109-07" "TCGA-AN-A0XO-01A-11R-A109-07"
 [19] "TCGA-AN-A0XP-01A-11R-A109-07" "TCGA-A2-A0YD-01A-11R-A109-07"
 [21] "TCGA-BH-A0W4-01A-11R-A109-07" "TCGA-B6-A0WZ-01A-11R-A109-07"
 [23] "TCGA-AN-A0XS-01A-22R-A109-07" "TCGA-BH-A0W5-01A-11R-A109-07"
 [25] "TCGA-AN-A0XT-01A-11R-A109-07" "TCGA-B6-A0X4-01A-11R-A109-07"
 [27] "TCGA-AO-A12G-01A-11R-A10J-07" "TCGA-AO-A125-01A-11R-A10J-07"
 [29] "TCGA-AO-A126-01A-11R-A10J-07" "TCGA-BH-A0B2-01A-11R-A10J-07"
 [31] "TCGA-A2-A0YI-01A-31R-A10J-07" "TCGA-E2-A10E-01A-21R-A10J-07"
 [33] "TCGA-AO-A12C-01A-11R-A10J-07" "TCGA-E2-A10F-01A-11R-A10J-07"
 [35] "TCGA-AO-A12E-01A-11R-A10J-07" "TCGA-E2-A106-01A-11R-A10J-07"
 [37] "TCGA-B6-A0X7-01A-11R-A10J-07" "TCGA-AN-A0XL-01A-11R-A10J-07"
 [39] "TCGA-AO-A03V-01A-11R-A115-07" "TCGA-BH-A0H5-01A-21R-A115-07"
 [41] "TCGA-A2-A04N-01A-11R-A115-07" "TCGA-AO-A12H-01A-11R-A115-07"
 [43] "TCGA-C8-A132-01A-31R-A115-07" "TCGA-D8-A141-01A-11R-A115-07"
 [45] "TCGA-E2-A15P-01A-11R-A115-07" "TCGA-BH-A0BP-01A-11R-A115-07"
 [47] "TCGA-A2-A0CS-01A-11R-A115-07" "TCGA-B6-A0IH-01A-11R-A115-07"
 [49] "TCGA-BH-A0BQ-01A-21R-A115-07" "TCGA-A2-A0CV-01A-31R-A115-07"
 [51] "TCGA-B6-A0WS-01A-11R-A115-07" "TCGA-BH-A0EA-01A-11R-A115-07"
 [53] "TCGA-B6-A0X0-01A-21R-A115-07" "TCGA-D8-A145-01A-11R-A115-07"
 [55] "TCGA-BH-A0B0-01A-21R-A115-07" "TCGA-A2-A0D3-01A-11R-A115-07"
 [57] "TCGA-BH-A0EI-01A-11R-A115-07" "TCGA-A1-A0SD-01A-11R-A115-07"
 [59] "TCGA-C8-A12N-01A-11R-A115-07" "TCGA-D8-A146-01A-31R-A115-07"
 [61] "TCGA-AO-A12A-01A-21R-A115-07" "TCGA-E2-A15D-01A-11R-A115-07"
 [63] "TCGA-BH-A0DE-01A-11R-A115-07" "TCGA-A2-A0EW-01A-21R-A115-07"
 [65] "TCGA-A8-A08T-01A-21R-A00Z-07" "TCGA-A7-A0CH-01A-21R-A00Z-07"
 [67] "TCGA-A8-A07J-01A-11R-A00Z-07" "TCGA-A8-A06Y-01A-21R-A00Z-07"
 [69] "TCGA-A8-A09A-01A-11R-A00Z-07" "TCGA-A8-A091-01A-11R-A00Z-07"
 [71] "TCGA-A8-A09B-01A-11R-A00Z-07" "TCGA-A7-A0CD-01A-11R-A00Z-07"
 [73] "TCGA-A7-A0DB-01A-11R-A00Z-07" "TCGA-A8-A09V-01A-11R-A034-07"
 [75] "TCGA-A8-A0A2-01A-11R-A034-07" "TCGA-A2-A0CP-01A-11R-A034-07"
 [77] "TCGA-A2-A0CQ-01A-21R-A034-07" "TCGA-A8-A06P-01A-11R-A00Z-07"
 [79] "TCGA-A8-A093-01A-11R-A00Z-07" "TCGA-A7-A0DC-01A-11R-A00Z-07"
 [81] "TCGA-AN-A046-01A-21R-A034-07" "TCGA-AN-A04A-01A-21R-A034-07"
 [83] "TCGA-A8-A07G-01A-11R-A034-07" "TCGA-BH-A0E7-01A-11R-A034-07"
 [85] "TCGA-A2-A0EM-01A-11R-A034-07" "TCGA-A2-A0EX-01A-21R-A034-07"
 [87] "TCGA-BH-A0EB-01A-11R-A034-07" "TCGA-BH-A0HO-01A-11R-A034-07"
```

```
 [89] "TCGA-B6-A0I5-01A-11R-A034-07" "TCGA-B6-A0I8-01A-11R-A034-07"
 [91] "TCGA-A2-A0EO-01A-11R-A034-07" "TCGA-A2-A0EV-01A-11R-A034-07"
 [93] "TCGA-AN-A0FS-01A-11R-A034-07" "TCGA-BH-A0HQ-01A-11R-A034-07"
 [95] "TCGA-AN-A0FN-01A-11R-A034-07" "TCGA-A7-A0CG-01A-12R-A056-07"
 [97] "TCGA-AO-A0J8-01A-21R-A034-07" "TCGA-B6-A0IP-01A-11R-A034-07"
 [99] "TCGA-AR-A0TR-01A-11R-A084-07" "TCGA-BH-A0DH-01A-11R-A084-07"
[101] "TCGA-AO-A0JG-01A-31R-A084-07" "TCGA-B6-A0RV-01A-11R-A084-07"
[103] "TCGA-BH-A0DQ-01A-11R-A084-07" "TCGA-B6-A0RN-01A-12R-A084-07"
[105] "TCGA-A2-A0EN-01A-13R-A084-07" "TCGA-B6-A0RO-01A-22R-A084-07"
[107] "TCGA-BH-A0HI-01A-11R-A084-07" "TCGA-A1-A0SE-01A-11R-A084-07"
[109] "TCGA-A2-A0SU-01A-11R-A084-07" "TCGA-A1-A0SH-01A-11R-A084-07"
[111] "TCGA-A2-A0T5-01A-21R-A084-07" "TCGA-B6-A0RP-01A-21R-A084-07"
[113] "TCGA-A2-A0T6-01A-11R-A084-07" "TCGA-BH-A0HP-01A-12R-A084-07"
[115] "TCGA-A2-A0SY-01A-31R-A084-07" "TCGA-BH-A18I-01A-11R-A12D-07"
[117] "TCGA-E2-A14Q-01A-11R-A12D-07" "TCGA-BH-A0BO-01A-23R-A12D-07"
[119] "TCGA-BH-A18S-01A-11R-A12D-07" "TCGA-E2-A15E-06A-11R-A12D-07"
[121] "TCGA-BH-A0DO-01B-11R-A12D-07" "TCGA-BH-A0DT-01A-21R-A12D-07"
[123] "TCGA-BH-A18M-01A-11R-A12D-07" "TCGA-E2-A15C-01A-31R-A12D-07"
[125] "TCGA-BH-A18N-01A-11R-A12D-07" "TCGA-C8-A133-01A-32R-A12D-07"
[127] "TCGA-E2-A15G-01A-11R-A12D-07" "TCGA-BH-A18H-01A-11R-A12D-07"
[129] "TCGA-E2-A153-01A-12R-A12D-07" "TCGA-E2-A15J-01A-11R-A12P-07"
[131] "TCGA-BH-A0AZ-01A-21R-A12P-07" "TCGA-E2-A1B4-01A-11R-A12P-07"
[133] "TCGA-BH-A0BS-01A-11R-A12P-07" "TCGA-BH-A0H3-01A-11R-A12P-07"
[135] "TCGA-AR-A1AN-01A-11R-A12P-07" "TCGA-BH-A0BT-01A-11R-A12P-07"
[137] "TCGA-BH-A0HA-01A-11R-A12P-07" "TCGA-BH-A1EU-01A-11R-A137-07"
[139] "TCGA-E2-A15I-01A-21R-A137-07" "TCGA-BH-A1EW-11B-33R-A137-07"
[141] "TCGA-BH-A1ET-01A-11R-A137-07" "TCGA-C8-A1HI-01A-11R-A137-07"
```

2. Getting the data with TCGAbiolinks R package

All the subject's data mentioned in section 1 was downloaded by means of the **TCGAbiolinks** R package, however, at the present this library had been greatly refactored, causing that this code does not work unless some files are present in your hard drive, these files are available upon request as they weigh too much. Below we show the code used to get both RDatas.

```
> ## Not run:
>
> library(TCGAbiolinks);
> R.Version()$version.string;
> # [1] "R version 3.2.3 (2015-12-10)"
> packageVersion("TCGAbiolinks");
> # [1] ‘1.0.10’
>
> query <- TCGAquery(tumor="BRCA");
> matSamples <- TCGAquery_integrate(query);
```

```
> # subjects in both microarray and RNAseq data
> matSamples["AgilentG4502A_07_3", "IlluminaHiSeq_RNASeq"];
> # [1] 495
>
> # we filter only microarray data
> geneExprSubjects <- TCGAquery(tumor="BRCA", platform="AgilentG4502A_07_3",
+     level=3);
> # we filter only RNAseq data
> rnaSeqSubjects <- TCGAquery(tumor="BRCA", platform="IlluminaHiSeq_RNASeq",
+     level=3);
> geneExprbarcodes <- geneExprSubjects$barcode;
> geneExprbarcodes <- strsplit(geneExprbarcodes, ",");
> geneExprbarcodes <- Reduce(union, geneExprbarcodes);
> rnaSeqbarcodes <- rnaSeqSubjects$barcode;
> rnaSeqbarcodes <- strsplit(rnaSeqbarcodes, ",");
> rnaSeqbarcodes <- Reduce(union, rnaSeqbarcodes);
> commonSubjects <- intersect(geneExprbarcodes, rnaSeqbarcodes);
> rm(geneExprbarcodes); rm(rnaSeqbarcodes);
> length(commonSubjects);
> # [1] 547
>
> # we filter microarray and RNAseq data (but just common subjects)
> geneExprSubjects <- TCGAquery(tumor="BRCA", platform="AgilentG4502A_07_3",
+     samples=commonSubjects, level=3);
> rnaSeqSubjects <- TCGAquery(tumor="BRCA", platform="IlluminaHiSeq_RNASeq",
+     samples=commonSubjects, level=3);
> #### this lines are the ones which are not working any more (TCGAdownload)
> # TCGAdownload(geneExprSubjects, path="geneExpr/", samples=commonSubjects);
> # TCGAdownload(rnaSeqSubjects,   path="rnaSeq/",   samples=commonSubjects,
> #     type="gene.quantification");
>
> ## However, we can provide you necessary files to skip the TCGAdownload step.
>
> ## type is any of:
> # RNASeq:             exon.quantification
> #                     spljxn.quantification
> #                     gene.quantification
> # genome_wide_snp_6:  hg18.seg
> #                     hg19.seg,nocnv_hg18.seg
> #                     nocnv_hg19.seg
>
> geneExpr <- TCGAprepare(geneExprSubjects, dir="geneExpr/");
> rnaSeq <- TCGAprepare(rnaSeqSubjects, dir="rnaSeq/",
+     type="gene.quantification");
> library(SummarizedExperiment);
> assays(geneExpr);
> # names(1): raw_counts
```

```
>
> # It would be a better way of conversion
> geneExpr <- head(assay(geneExpr, "raw_counts"), n=nrow(geneExpr));
> assays(rnaSeq);
> # names(3): raw_counts median_length_normalized RPKM
> rnaSeq_raw <- head(assay(rnaSeq, "raw_counts"), n=nrow(rnaSeq));
> rnaSeq_medianNorm <- head(assay(rnaSeq, "median_length_normalized"),
+     n=nrow(rnaSeq));
> rnaSeq_rpkm <- head(assay(rnaSeq, "RPKM"), n=nrow(rnaSeq));
> ## checking if we have the same subjects in every experiment
> stopifnot(all(colnames(geneExpr) %in% colnames(rnaSeq_raw)));
> stopifnot(all(colnames(rnaSeq_raw) %in% colnames(rnaSeq_medianNorm)));
> stopifnot(all(colnames(rnaSeq_medianNorm) %in% colnames(rnaSeq_rpkm)));
> stopifnot(all(colnames(rnaSeq_rpkm) %in% colnames(geneExpr)));
> mapping <- do.call(rbind, strsplit(rownames(rnaSeq_raw), "|", fixed=!F));
> colnames(mapping) <- c("Symbol", "Entrez");
> #### Now let's get subjects subtypes
>
> library(genefu);
> rnaSeq <- rnaSeq_rpkm;
> rm(rnaSeq_rpkm);
> ## Also request this file!
> pam50Annot <- read.csv("pam50_annotation.txt",sep="\t");
> library(limma);
> dim(geneExpr);
> geneExpr <- avereps(geneExpr);
> dim(geneExpr);
> rownames(rnaSeq) <- mapping[, "Symbol" ];
> dim(rnaSeq);
> rnaSeq <- rnaSeq[ mapping[, "Symbol" ] != "?" , ];
> dim(rnaSeq);
> rnaSeq <- avereps(rnaSeq);
> dim(rnaSeq);
> geneExpr <- geneExpr[as.character(pam50Annot$GeneName),, drop=F];
> dim(geneExpr);
> rnaSeq <- rnaSeq[as.character(pam50Annot$GeneName),, drop=F];
> dim(rnaSeq);
> rnaSeq <- log(rnaSeq);
> pam50Annot <- pam50Annot[,c("GeneName", "EntrezGene")];
> colnames(pam50Annot) <- c("probe", "EntrezGene.ID");
> pam50Annot$probe <- as.character(pam50Annot$probe);
> ## get subtypes
> dataset <- apply(geneExpr, 1, as.numeric);
> rownames(dataset) <- colnames(geneExpr);
> subtypesGeneExpr <- intrinsic.cluster.predict(sbt.model=pam50.scale,
+     data=dataset, annot=pam50Annot, do.mapping=!F, do.prediction.strength=!F,
+     verbose=!F);
```

```
> ## get subtypes
> dataset <- apply(rnaSeq, 1, as.numeric);
> rownames(dataset) <- colnames(rnaSeq);
> subtypesRnaSeq <- intrinsic.cluster.predict(sbt.model=pam50.scale,
+     data=dataset, annot=pam50Annot, do.mapping=!F, do.prediction.strength=!F,
+     verbose=!F);
> table(subtypesGeneExpr$subtype);
> #  Basal   Her2   LumA   LumB Normal
> #    101     77    150    157     62
>
> table(subtypesRnaSeq$subtype);
> #  Basal   Her2   LumA   LumB Normal
> #    101     81    165    137     63
>
> subtypesGeneExpr <- subtypesGeneExpr$subtype;
> subtypesRnaSeq <- subtypesRnaSeq$subtype[names(subtypesGeneExpr)];
> ## how many subjects got the same subtype between microarray and RNAseq data
> concSubtypes <- table(subtypesGeneExpr, subtypesRnaSeq);
> concSubtypes;
> #          Basal Her2 LumA LumB Normal
> #   Basal     95    2    1    2      1
> #   Her2       0   72    0    4      1
> #   LumA       1    0  142    4      3
> #   LumB       3    7   19  127      1
> #   Normal     2    0    3    0     57
> sum(diag(concSubtypes)) / sum(concSubtypes);
> # [1] 0.9012797 # 90% of concordant subjects
>
> stopifnot(all(names(subtypesGeneExpr) == names(subtypesRnaSeq)));
> ## I am going to use the subjects that got the same classification in both
> subtypes <- subtypesGeneExpr[subtypesGeneExpr == subtypesRnaSeq];
> length(subtypes);
> # [1] 493
>
> #### Now just translate GeneSymbols to EntrezGene IDs
>
> ## Also request this file!
> annotAgi <- read.csv("AgilentG4502A_07_3.csv", sep="|");
> geneExprSymbol <- rownames(geneExpr);
> # we first search into Agilent annotation file
> geneExprEntrez <- annotAgi[ match(geneExprSymbol, annotAgi[, "Symbol"]),
+     "Entrez" ];
> sum(is.na(geneExprEntrez));
> # [1] 796
> # then we look into the mapping given by RNASeq TCGA data
> geneExprEntrez[ is.na(geneExprEntrez) ] <- mapping[ match(geneExprSymbol[
+     is.na(geneExprEntrez) ], mapping[, "Symbol"]), "Entrez" ];
```

```
> sum(is.na(geneExprEntrez));
> # [1] 772
>
> geneExpr <- geneExpr[ !is.na(geneExprEntrez), ];
> rownames(geneExpr) <- geneExprEntrez[ !is.na(geneExprEntrez) ];
> dim(geneExpr);
> geneExpr <- avereps(geneExpr);
> dim(geneExpr);
> rownames(rnaSeq) <- do.call(rbind, strsplit(rownames(rnaSeq), "|",
+     fixed=!F))[,2];
> dim(rnaSeq);
> rnaSeq <- avereps(rnaSeq);
> dim(rnaSeq);
> load("rnaSeq_raw.RData");
> rownames(rnaSeq_raw) <- do.call(rbind, strsplit(rownames(rnaSeq_raw), "|",
+     fixed=!F))[,2];
> dim(rnaSeq_raw);
> rnaSeq_raw <- avereps(rnaSeq_raw);
> dim(rnaSeq_raw);
> #### And keep only Basal and Luminal A subjects
> rnaSeq_raw <- rnaSeq_raw[, names(subtypes)[subtypes %in% c("Basal", "LumA")] ];
> geneExpr <- geneExpr[, names(subtypes)[subtypes %in% c("Basal", "LumA")] ];
> subtypes <- subtypes[subtypes %in% c("Basal", "LumA")];
> ## And these are the two data objects used.
> tcgaRNAseqData <- list(rnaSeq=rnaSeq_raw, subtypes=subtypes);
> tcgaMAdata <- list(geneExpr=geneExpr, subtypes=subtypes);
> ## End(Not run)
```

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/