curatedCRCData

Princy Parsana, Markus Riester, Curtis Huttenhower, Levi Waldron

2013

Contents

1 curatedCRCData: Clinically Annotated Data for the colorectal Cancer Transcriptome . . . 1

2 Load data sets . . . 1

3 Load datasets based on rules . . . 2

4 Non-unique gene symbols . . . 6

A Available Clinical Characteristics . . . 7

B Summarizing the List of ExpressionSets . . . 7

C For non-R users . . . 10

D Session Info . . . 10

1 curatedCRCData: Clinically Annotated Data for the colorectal Cancer Transcriptome

This package represents a manually curated data collection for gene expression meta-analysis of patients with colorectal cancer. This resource provides uniformly prepared microarray data with curated and documented clinical metadata. It allows a computational user to efficiently identify studies and patient subgroups of interest for analysis and to run such analyses immediately without the challenges posed by harmonizing heterogeneous microarray technologies, study designs, expression data processing methods, and clinical data formats.

In this vignette, we give a short tour of the package and will show how to use it efficiently.

2 Load data sets

Loading a single dataset is very easy. First we load the package:

```
> library(curatedCRCData)
```

To get a listing of all the datasets, use the `data` function:

```
> data(package="curatedCRCData")
```

Now to load a single dataset, we use the `data` function again:

```
> data(TCGA.COAD_eset)
> TCGA.COAD_eset

ExpressionSet (storageMode: lockedEnvironment)
assayData: 17814 features, 130 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: TCGA.AA.3520 TCGA.AA.3532 ... TCGA.A6.2685 (130 total)
  varLabels: unique_patient_ID alt_sample_name ...
    uncurated_author_metadata (59 total)
  varMetadata: labelDescription
featureData
  featureNames: 15E1.2 2'-PDE ... ZZZ3 (17814 total)
  fvarLabels: probeset gene
  fvarMetadata: labelDescription
experimentData: use 'experimentData(object)'
  pubMedIds: 22810696
Annotation: agilent-014850 whole human genome microarray 4x44k g4112f
```

The datasets are provided as Bioconductor `ExpressionSet` objects and we refer to the Bioconductor documentation for users unfamiliar with this data structure.

3 Load datasets based on rules

For a meta-analysis, we typically want to filter datasets and patients to get a population of patients we are interested in. We provide a short but powerful R script that does the filtering and provides the data as a list of `ExpressionSet` objects. One can use this script within R by first sourcing a config file which specifies the filters, like the minimum numbers of patients in each dataset. It is also possible to filter samples by annotation, for example to remove early stage and normal samples.

```
> source(system.file("extdata",
+ "patientselection_all.config",package="curatedCRCData"))
> ls()

[1] "TCGA.COAD_eset"       "keep.common.only"     "meta.required"
[4] "min.number.of.events" "min.sample.size"      "quantile.cutoff"
[7] "rescale"              "strict.checking"
```

See what the values of these variables we have loaded are. The variable names are fairly descriptive, but note that "rule.1" is a character vector of length 2, where the first entry is the name of a clinical data variable, and the second entry is a Regular Expression providing a requirement for that variable. Any number of rules can be added, with increasing identifiers, e.g. "rule.2", "rule.3", etc.

Here strict.checking is FALSE, meaning that samples not annotated for the variables in these rules are allowed to pass the filter. If strict.checking == TRUE, samples missing this annotation will be removed.

```
> sapply(ls(), get)

$TCGA.COAD_eset
ExpressionSet (storageMode: lockedEnvironment)
assayData: 17814 features, 130 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: TCGA.AA.3520 TCGA.AA.3532 ... TCGA.A6.2685 (130 total)
  varLabels: unique_patient_ID alt_sample_name ...
    uncurated_author_metadata (59 total)
  varMetadata: labelDescription
featureData
  featureNames: 15E1.2 2'-PDE ... ZZZ3 (17814 total)
  fvarLabels: probeset gene
  fvarMetadata: labelDescription
experimentData: use 'experimentData(object)'
  pubMedIds: 22810696
Annotation: agilent-014850 whole human genome microarray 4x44k g4112f

$keep.common.only
[1] FALSE

$meta.required
NULL

$min.number.of.events
[1] 0

$min.sample.size
[1] 1

$quantile.cutoff
[1] 0

$rescale
[1] FALSE

$strict.checking
[1] FALSE
```

Now that we have defined the sample filter, we create a list of `ExpressionSets` by sourcing the `createEsetList.R` file:

```
> source(system.file("extdata", "createEsetList.R", package =
+ "curatedCRCData"))

2023-04-27 10:20:03.744182 INFO::Inside script createEsetList.R - inputArgs =
2023-04-27 10:20:03.755735 INFO::None provided
```

```
2023-04-27 10:20:03.829886 INFO::Loading curatedCRCData 2.33.0
2023-04-27 10:20:47.750672 INFO::Clean up the esets.
2023-04-27 10:20:47.842632 INFO::including GSE11237_eset
2023-04-27 10:20:47.868183 INFO::including GSE12225.GPL3676_eset
2023-04-27 10:20:47.901312 INFO::including GSE12945_eset
2023-04-27 10:20:47.939424 INFO::including GSE13067_eset
2023-04-27 10:20:48.019858 INFO::including GSE13294_eset
2023-04-27 10:20:48.503725 INFO::including GSE14095_eset
2023-04-27 10:20:48.609235 INFO::including GSE14333_eset
2023-04-27 10:20:48.683079 INFO::including GSE16125.GPL5175_eset
2023-04-27 10:20:48.751227 INFO::including GSE17536_eset
2023-04-27 10:20:48.8158 INFO::including GSE17537_eset
2023-04-27 10:20:48.901021 INFO::including GSE17538.GPL570_eset
2023-04-27 10:20:49.358536 INFO::including GSE18105_eset
2023-04-27 10:20:49.477133 INFO::including GSE2109_eset
2023-04-27 10:20:49.597034 INFO::including GSE21510_eset
2023-04-27 10:20:49.668213 INFO::including GSE21815_eset
2023-04-27 10:20:49.722639 INFO::including GSE24549.GPL5175_eset
2023-04-27 10:20:49.755283 INFO::including GSE24550.GPL5175_eset
2023-04-27 10:20:49.784341 INFO::including GSE2630_eset
2023-04-27 10:20:49.811506 INFO::including GSE26682.GPL570_eset
2023-04-27 10:20:49.851462 INFO::including GSE26682.GPL96_eset
2023-04-27 10:20:50.285073 INFO::including GSE26906_eset
2023-04-27 10:20:50.314907 INFO::including GSE27544_eset
2023-04-27 10:20:50.359969 INFO::including GSE28702_eset
2023-04-27 10:20:50.400672 INFO::including GSE3294_eset
2023-04-27 10:20:50.442928 INFO::including GSE33113_eset
2023-04-27 10:20:50.57519 INFO::including GSE39582_eset
2023-04-27 10:20:50.697425 INFO::including GSE3964_eset
2023-04-27 10:20:50.712796 INFO::including GSE4045_eset
2023-04-27 10:20:50.735547 INFO::including GSE4526_eset
2023-04-27 10:20:50.76142 INFO::including GSE45270_eset
2023-04-27 10:20:50.817921 INFO::including TCGA.COAD_eset
2023-04-27 10:20:50.861881 INFO::including TCGA.READ_eset
2023-04-27 10:20:50.885647 INFO::including TCGA.RNASeqV2.READ_eset
2023-04-27 10:20:50.938747 INFO::including TCGA.RNASeqV2_eset
2023-04-27 10:20:51.677455 INFO::Ids with missing data: GSE2630_eset, GSE3294_eset, TCGA.COAD_eset, TCGA.REAL
```

It is also possible to run the script from the command line and then load the R data file within R:

```
R --vanilla "--args patientselection.config crc.eset.rda tmp.log"  < createEsetList.R
```

Now we have 34 datasets with samples that passed our filter in a list of `ExpressionSets` called `esets`:

```
> names(esets)

 [1] "GSE11237_eset"           "GSE12225.GPL3676_eset"
 [3] "GSE12945_eset"           "GSE13067_eset"
 [5] "GSE13294_eset"           "GSE14095_eset"
 [7] "GSE14333_eset"           "GSE16125.GPL5175_eset"
 [9] "GSE17536_eset"           "GSE17537_eset"
```

```
 [11] "GSE17538.GPL570_eset"    "GSE18105_eset"
 [13] "GSE2109_eset"            "GSE21510_eset"
 [15] "GSE21815_eset"           "GSE24549.GPL5175_eset"
 [17] "GSE24550.GPL5175_eset"   "GSE2630_eset"
 [19] "GSE26682.GPL570_eset"    "GSE26682.GPL96_eset"
 [21] "GSE26906_eset"           "GSE27544_eset"
 [23] "GSE28702_eset"           "GSE3294_eset"
 [25] "GSE33113_eset"           "GSE39582_eset"
 [27] "GSE3964_eset"            "GSE4045_eset"
 [29] "GSE4526_eset"            "GSE45270_eset"
 [31] "TCGA.COAD_eset"          "TCGA.READ_eset"
 [33] "TCGA.RNASeqV2.READ_eset" "TCGA.RNASeqV2_eset"
```

4 Non-unique gene symbols

In the standard version of curatedCRCData (the version available on Bioconductor), we collapse manufacturer probesets to official HGNC symbols using the Biomart database. Some probesets are mapped to multiple HGNC symbols in this database. For these probesets, we provide all the symbols. For example `220159_at` maps to *ABCA11P* and *ZNF721* and we provide `ABCA11P///ZNF721` as probeset name. If you have an array of gene symbols for which you want to access the expression data, "ABCA11P" would not be found in curatedCRCData in this example. The following function will create a new ExpressionSet in which both *ZNF721* and *ABCA11P* are features with identical expression data:

```
> expandProbesets <- function (eset, sep = "///")
+ {
+     x <- lapply(featureNames(eset), function(x) strsplit(x, sep)[[1]])
+     eset <- eset[order(sapply(x, length)), ]
+     x <- lapply(featureNames(eset), function(x) strsplit(x, sep)[[1]])
+     idx <- unlist(sapply(1:length(x), function(i) rep(i, length(x[[i]]))))
+     xx <- !duplicated(unlist(x))
+     idx <- idx[xx]
+     x <- unlist(x)[xx]
+     eset <- eset[idx, ]
+     featureNames(eset) <- x
+     eset
+ }
> X <- TCGA.COAD_eset[head(grep("AA", featureNames(TCGA.COAD_eset))),]
> exprs(X)[,1:3]

        TCGA.AA.3520 TCGA.AA.3532 TCGA.AA.3553
AAAS         -0.72125     -1.51150     -1.01250
AACS          0.02225      0.82375     -0.08500
AADAC         0.02775     -0.42900      1.12525
AADACL1       1.06600      1.85550      0.92800
AADACL2       0.08750     -0.61825     -0.47525
AADACL3       0.38100      0.31100      0.33950

> exprs(expandProbesets(X))[,1:3]

        TCGA.AA.3520 TCGA.AA.3532 TCGA.AA.3553
AAAS         -0.72125     -1.51150     -1.01250
AACS          0.02225      0.82375     -0.08500
AADAC         0.02775     -0.42900      1.12525
AADACL1       1.06600      1.85550      0.92800
AADACL2       0.08750     -0.61825     -0.47525
AADACL3       0.38100      0.31100      0.33950
```

A Available Clinical Characteristics

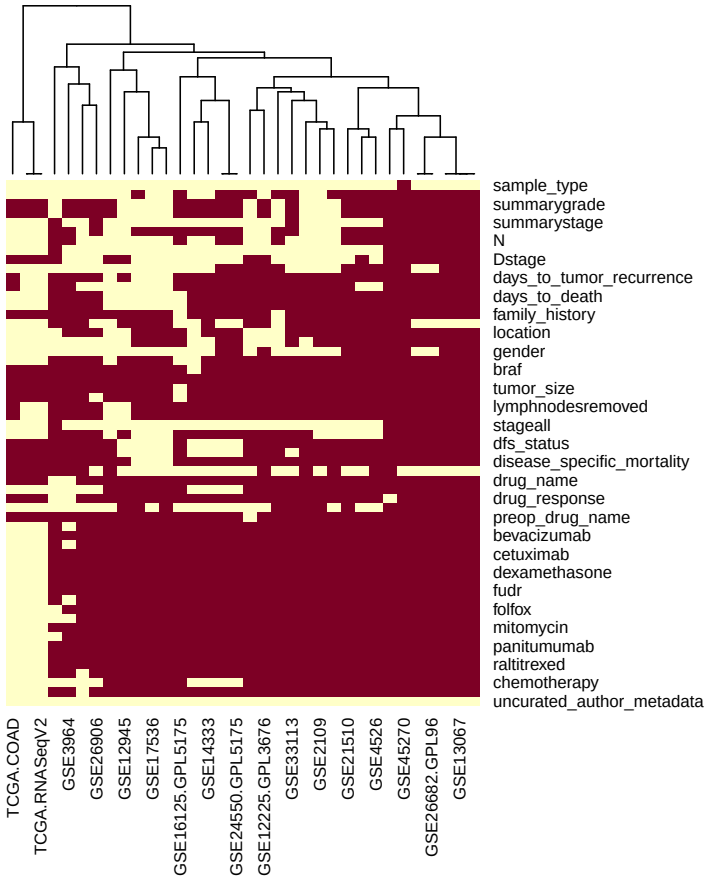

Figure 1: Available clinical annotation. This heatmap visualizes for each curated clinical characteristic (rows) the availability in each dataset (columns). Red indicates that the corresponding characteristic is available for at least one sample in the dataset.

B Summarizing the List of ExpressionSets

This example provides a table summarizing the datasets being used, and is useful when publishing analyses based on curatedCRCData. First, define some useful functions for this purpose:

```
> source(system.file("extdata", "summarizeEsets.R", package =
+     "curatedCRCData"))
```

Optionally write this table to file, for example (replace myfile <- tempfile() with something like myfile <- "nicetable.csv")

```
> (myfile <- tempfile())

[1] "/tmp/Rtmpt0EnID/file2984f5fbcea36"
```

```
> write.table(summary.table, file=myfile, row.names=FALSE, quote=TRUE, sep=",")
```

	Study	PMID	samples
GSE11237	Auman, Mcleod 2008	18653328	3
GSE12225.GPL3676	Lips, Morreau 2008	18959792	2
GSE12945	Staub, Rosenthal 2009	19399471	2
GSE13067	Jorissen and Sieber 2008	19088021	4
GSE13294	Jorissen and Sieber 2008	19088021	155
GSE14095	Watanabe, Hashimoto 2008	21680303	189
GSE14333	Jorissen and Sieber 2008	19996206	290
GSE16125.GPL5175	Reid, Pierotti 2009	19672874	36
GSE17536	Smith JJ,??Beauchamp RD 2009	19914252	177
GSE17537	Smith JJ,??Beauchamp RD 2009	19914252	55
GSE17538.GPL570	Smith JJ,??Beauchamp RD 2009	19914252	232
GSE18105	Matsuyama, Sugihara 2009	20162577	111
GSE2109	expO, IGC 2005	PMID unknown	427
GSE21510	Tsukamoto, Sugihara 2010	21270110	148
GSE21815	Mori M,??Mimori K,??Yokobori T 2010	21862635	141
GSE24549.GPL5175	Sveen A,????esen TH,??Rognum TO,??Lothe RA,??Skotheim RI 2011	21619627	83
GSE24550.GPL5175	Sveen A,????esen TH,??Rognum TO,??Lothe RA,??Skotheim RI 2011	21619627	90
GSE2630	Bandr??E,??Malumbres R,??Cubedo E,??Sola J,??Garc??Foncillas J,??Labarga A 2005	17390049	16
GSE26682.GPL570	Vilar E,??Morgan MA 2011	21300766	156
GSE26682.GPL96	Vilar E,??Morgan MA 2011	21300766	155
GSE26906	Olschwang S 2011	22496922	90
GSE27544	Bernal M,??Garc??Alcalde F,??Concha ????Blanco A,??Garrido F,??Ruiz-Cabello F 2011	PMID unknown	22
GSE28702	Tsuji 2011	22095227	83
GSE3294	Bianchini 2005	16773188	24
GSE33113	Medema JP,??Tanis PJ 2011	22496204	96
GSE39582	Marisa, Boige 2012	23700391	566
GSE3964	Graudens, Imbeaud 2006	16542501	29
GSE4045	Laiho, Aaltonen 2007	16819509	37
GSE4526	Watanabe T,??Kobunai T,??Toda E,??Oka T 2006	19016304	36
GSE45270	Medema JP,??de Sousa E Melo F,??Vermeulen L,??Jansen M,??Dekker E,??Van Noesel C,??Fessler E 2013	PMID unknown	13
TCGA.COAD	The Cancer Genome Atlas Network 2012	22810696	130
TCGA.READ	The Cancer Genome Atlas Network 2012	22810696	51
TCGA.RNASeqV2.READ	The Cancer Genome Atlas Network 2012	22810696	6
TCGA.RNASeqV2	The Cancer Genome Atlas Network 2012	22810696	195

Table 1: Datasets provided by curatedCRCData.

C For non-R users

If you are not doing your analysis in R, and just want to get some data you have identified from the curatedCRCData manual, here is a simple way to do it. For one dataset:

```
> library(curatedCRCData)
> library(affy)
> data(TCGA.COAD_eset)
> write.csv(exprs(TCGA.COAD_eset), file="TCGA.COAD_eset_exprs.csv")
> write.csv(pData(TCGA.COAD_eset), file="TCGA.COAD_eset_clindata.csv")
```

Or for several datasets:

```
> data.to.fetch <- c("TCGA.COAD_eset", "GSE37317_eset")
> for (onedata in data.to.fetch){
+     print(paste("Fetching", onedata))
+     data(list=onedata)
+     write.csv(exprs(get(onedata)), file=paste(onedata, "_exprs.csv", sep=""))
+     write.csv(pData(get(onedata)), file=paste(onedata, "_clindata.csv", sep=""))
+ }
```

D Session Info

- R version 4.3.0 RC (2023-04-18 r84287), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_GB`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Time zone: `America/New_York`
- TZcode source: `system (glibc)`
- Running under: `Ubuntu 22.04.2 LTS`
- Matrix products: default
- BLAS: `/home/biocbuild/bbs-3.18-bioc/R/lib/libRblas.so`
- LAPACK: `/usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: Biobase 2.61.0, BiocGenerics 0.47.0, BiocParallel 1.35.0, affy 1.79.0, curatedCRCData 2.33.0, genefilter 1.83.0, logging 0.10-108, mgcv 1.8-42, nlme 3.1-162, survival 3.5-5, sva 3.49.0, xtable 1.8-4
- Loaded via a namespace (and not attached): AnnotationDbi 1.63.0, BiocManager 1.30.20, BiocStyle 2.29.0, Biostrings 2.69.0, DBI 1.1.3, GenomeInfoDb 1.37.0, GenomeInfoDbData 1.2.10, IRanges 2.35.0, KEGGREST 1.41.0, Matrix 1.5-4, MatrixGenerics 1.13.0, R6 2.5.1, RCurl 1.98-1.12, RSQLite 2.3.1, Rcpp 1.0.10, S4Vectors 0.39.0, XML 3.99-0.14, XVector 0.41.0, affyio 1.71.0, annotate 1.79.0, bit 4.0.5, bit64 4.0.5, bitops 1.0-7, blob 1.2.4, cachem 1.0.7, cli 3.6.1, codetools 0.2-19, compiler 4.3.0, crayon 1.5.2, digest 0.6.31,

edgeR 3.43.0, evaluate 0.20, fastmap 1.1.1, grid 4.3.0, htmltools 0.5.5, httr 1.4.5, knitr 1.42, lattice 0.21-8, limma 3.57.0, locfit 1.5-9.7, matrixStats 0.63.0, memoise 2.0.1, parallel 4.3.0, png 0.1-8, preprocessCore 1.63.0, rlang 1.1.0, rmarkdown 2.21, splines 4.3.0, stats4 4.3.0, tools 4.3.0, vctrs 0.6.2, xfun 0.39, yaml 2.3.7, zlibbioc 1.47.0