curatedCRCData

Princy Parsana, Markus Riester, Curtis Huttenhower, Levi Waldron

2013

Contents

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 ... moltherapy (60 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 ... moltherapy (60 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"))
```

```
2014-10-18 07:49:59 INFO::Inside script createEsetList.R - inputArgs =
2014-10-18 07:49:59 INFO::
2014-10-18 07:49:59 INFO::Loading curatedCRCData 1.1.2
2014-10-18 07:50:59 INFO::Clean up the esets.
2014-10-18 07:50:59 INFO::including GSE11237_eset
2014-10-18 07:50:59 INFO::including GSE12225.GPL3676_eset
2014-10-18 07:50:59 INFO::including GSE12945_eset
2014-10-18 07:50:59 INFO::including GSE13067_eset
2014-10-18 07:50:59 INFO::including GSE13294_eset
2014-10-18 07:51:00 INFO::including GSE14095_eset
2014-10-18 07:51:01 INFO::including GSE14333_eset
2014-10-18 07:51:01 INFO::including GSE16125.GPL5175_eset
2014-10-18 07:51:01 INFO::including GSE17536_eset
2014-10-18 07:51:01 INFO::including GSE17537_eset
2014-10-18 07:51:01 INFO::including GSE17538.GPL570_eset
2014-10-18 07:51:02 INFO::including GSE18105_eset
2014-10-18 07:51:02 INFO::including GSE2109_eset
2014-10-18 07:51:02 INFO::including GSE21510_eset
2014-10-18 07:51:03 INFO::including GSE21815_eset
2014-10-18 07:51:03 INFO::including GSE24549.GPL5175_eset
```

```
2014-10-18 07:51:03 INFO::including GSE24550.GPL5175_eset
2014-10-18 07:51:03 INFO::including GSE2630_eset
2014-10-18 07:51:03 INFO::including GSE26682.GPL570_eset
2014-10-18 07:51:03 INFO::including GSE26682.GPL96_eset
2014-10-18 07:51:03 INFO::including GSE26906_eset
2014-10-18 07:51:03 INFO::including GSE27544_eset
2014-10-18 07:51:03 INFO::including GSE28702_eset
2014-10-18 07:51:03 INFO::including GSE3294_eset
2014-10-18 07:51:04 INFO::including GSE33113_eset
2014-10-18 07:51:04 INFO::including GSE39582_eset
2014-10-18 07:51:05 INFO::including GSE3964_eset
2014-10-18 07:51:05 INFO::including GSE4045_eset
2014-10-18 07:51:05 INFO::including GSE4526_eset
2014-10-18 07:51:05 INFO::including GSE45270_eset
2014-10-18 07:51:05 INFO::including TCGA.COAD_eset
2014-10-18 07:51:05 INFO::including TCGA.READ_eset
2014-10-18 07:51:05 INFO::including TCGA.RNASeqV2.READ_eset
2014-10-18 07:51:05 INFO::including TCGA.RNASeqV2_eset
2014-10-18 07:51:06 INFO::Ids with missing data: GSE3294_eset, TCGA.COAD_eset, TCGA.READ_eset
```

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"   "GSE12945_eset"
 [4] "GSE13067_eset"           "GSE13294_eset"           "GSE14095_eset"
 [7] "GSE14333_eset"           "GSE16125.GPL5175_eset"   "GSE17536_eset"
[10] "GSE17537_eset"           "GSE17538.GPL570_eset"    "GSE18105_eset"
[13] "GSE2109_eset"            "GSE21510_eset"           "GSE21815_eset"
[16] "GSE24549.GPL5175_eset"   "GSE24550.GPL5175_eset"   "GSE2630_eset"
[19] "GSE26682.GPL570_eset"    "GSE26682.GPL96_eset"     "GSE26906_eset"
[22] "GSE27544_eset"           "GSE28702_eset"           "GSE3294_eset"
[25] "GSE33113_eset"           "GSE39582_eset"           "GSE3964_eset"
[28] "GSE4045_eset"            "GSE4526_eset"            "GSE45270_eset"
[31] "TCGA.COAD_eset"          "TCGA.READ_eset"          "TCGA.RNASeqV2.READ_eset"
[34] "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

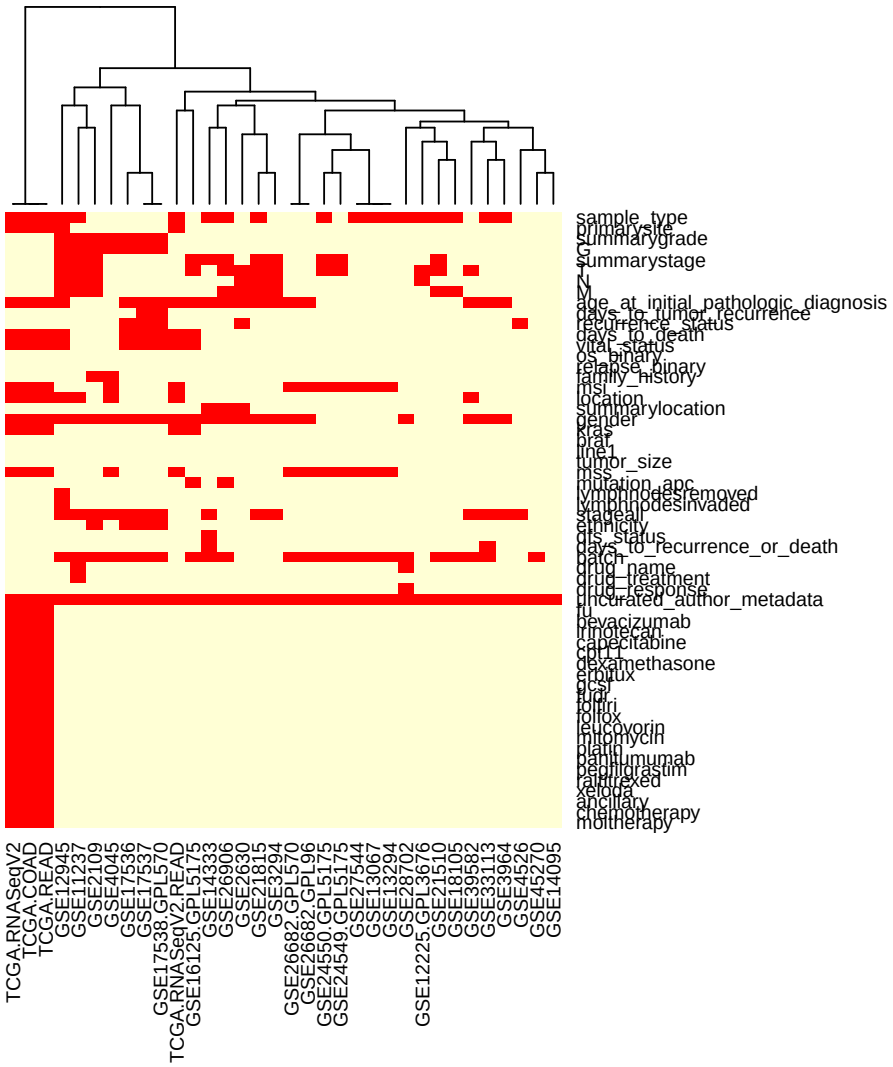

Figure 1: Available clinical annotation. This heatmap visualizes for each curated clinical characteristic (rows) the availability in each dataset (columns). Red indicates that the corresponding characteristic is available for at least one sample in the dataset.

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/RtmpU4fAzR/file7897422f4f2"
```

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

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 3.1.1 Patched (2014-09-25 r66681), x86_64-unknown-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8, LC_NUMERIC=C, LC_TIME=en_US.UTF-8, 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
- Base packages: base, datasets, grDevices, graphics, methods, parallel, splines, stats, utils
- Other packages: Biobase 2.26.0, BiocGenerics 0.12.0, curatedCRCData 1.1.2, genefilter 1.48.1, logging 0.7-103, mgcv 1.8-3, nlme 3.1-118, survival 2.37-7, sva 3.12.0, xtable 1.7-4
- Loaded via a namespace (and not attached): AnnotationDbi 1.28.0, BiocStyle 1.4.1, DBI 0.3.1, GenomeInfoDb 1.2.0, IRanges 2.0.0, Matrix 1.1-4, RSQLite 0.11.4, S4Vectors 0.4.0, XML 3.98-1.1, annotate 1.44.0, grid 3.1.1, lattice 0.20-29, stats4 3.1.1, tools 3.1.1

	PMID	N samples	stage	histology	Platform
GSE11237	18653328	23	4/19/0	0/0/0/0/0/0/23	Affymetrix HG_U95Av2
GSE12225.GPL3676	18959792	79	0/0/79	0/0/0/0/0/0/79	
GSE12945	19399471	62	16/46/0	0/0/0/0/0/0/62	Affymetrix HG-U133A
GSE13067	19088021	74	0/0/74	0/0/0/0/0/0/74	Affymetrix HG-U133Plus2
GSE13294	19088021	155	0/0/155	0/0/0/0/0/0/155	Affymetrix HG-U133Plus2
GSE14095	21680303	189	0/0/189	0/0/0/0/0/0/189	Affymetrix HG-U133Plus2
GSE14333	19996206	290	138/152/0	0/0/0/0/0/0/290	Affymetrix HG-U133Plus2
GSE16125.GPL5175	19672874	36	6/27/3	0/0/0/0/0/0/36	Affymetrix HuEx-1_0-st
GSE17536	19914252	177	0/0/177	0/0/0/0/0/0/177	Affymetrix HG-U133Plus2
GSE17537	19914252	55	0/0/55	0/0/0/0/0/0/55	Affymetrix HG-U133Plus2
GSE17538.GPL570	19914252	238	0/0/238	0/0/0/0/0/0/238	Affymetrix HG-U133Plus2
GSE18105	20162577	111	0/0/111	0/0/0/0/0/0/111	Affymetrix HG-U133Plus2
GSE2109	PMID unknown	428	166/168/94	0/0/0/0/0/0/428	Affymetrix HG-U133Plus2
GSE21510	21270110	148	73/74/1	0/0/0/0/0/0/148	Affymetrix HG-U133Plus2
GSE21815	21862635	141	14/52/75	0/0/0/0/0/0/141	Agilent G4112F
GSE24549.GPL5175	21619627	83	46/37/0	0/0/0/0/0/0/83	Affymetrix HuEx-1_0-st
GSE24550.GPL5175	21619627	90	44/33/13	0/0/0/0/0/0/90	Affymetrix HuEx-1_0-st
GSE2630	17390049	16	0/0/16	0/0/0/0/0/0/16	
GSE26682.GPL570	21300766	176	0/0/176	0/0/0/0/0/0/176	Affymetrix HG-U133Plus2
GSE26682.GPL96	21300766	155	0/0/155	0/0/0/0/0/0/155	Affymetrix HG-U133A
GSE26906	22496922	90	90/0/0	0/0/0/0/0/0/90	Affymetrix HG-U133Plus2
GSE27544	PMID unknown	22	0/0/22	0/0/0/0/0/0/22	Affymetrix HT HG-U133+ PM
GSE28702	22095227	59	0/0/59	0/0/0/0/0/0/59	Affymetrix HG-U133Plus2
GSE3294	16773188	24	4/20/0	0/0/0/0/0/0/24	UHN SS-Human 19Kv7
GSE33113	22496204	96	0/0/96	0/0/0/0/0/0/96	Affymetrix HG-U133Plus2
GSE39582	23700391	566	0/0/566	0/0/0/0/0/0/566	Affymetrix HG-U133Plus2
GSE3964	16542501	15	0/0/15	0/0/0/0/0/0/15	
GSE4045	16819509	37	0/0/37	0/0/0/0/0/0/37	Affymetrix HG-U133A
GSE4526	19016304	36	0/0/36	0/0/0/0/0/0/36	Affymetrix HG-U133Plus2
GSE45270	PMID unknown	13	0/0/13	0/0/0/0/0/0/13	Affymetrix HG-U133Plus2
TCGA.COAD	22810696	130	0/0/130	0/0/0/0/0/0/130	Agilent G4502A-07-3
TCGA.READ	22810696	51	0/0/51	0/0/0/0/0/0/51	Agilent G4502A-07-3
TCGA.RNASeqV2.READ	22810696	6	0/0/6	0/0/0/0/0/0/6	
TCGA.RNASeqV2	22810696	195	0/0/195	0/0/0/0/0/0/195	

Table 1: Datasets provided by curatedCRCData.