

# curatedCRCDData

Princy Parsana, Markus Riester, Curtis Huttenhower, Levi Waldron

2013

## Contents

---

<b>1</b>	<b>curatedCRCDData: Clinically Annotated Data for the colorectal Cancer Transcriptome</b>	<b>1</b>
<b>2</b>	<b>Load data sets</b>	<b>1</b>
<b>3</b>	<b>Load datasets based on rules</b>	<b>2</b>
<b>4</b>	<b>Non-unique gene symbols</b>	<b>5</b>
<b>A</b>	<b>Available Clinical Characteristics</b>	<b>6</b>
<b>B</b>	<b>Summarizing the List of ExpressionSets</b>	<b>6</b>
<b>C</b>	<b>For non-R users</b>	<b>7</b>
<b>D</b>	<b>Session Info</b>	<b>7</b>

## 1 curatedCRCDData: 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(curatedCRCDData)
```

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

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

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="curatedCRCDData"))
> 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"))

```

```

2015-10-15 13:10:26 INFO::Inside script createEsetList.R - inputArgs =
2015-10-15 13:10:26 INFO::
2015-10-15 13:10:26 INFO::Loading curatedCRCData 2.2.0
2015-10-15 13:11:18 INFO::Clean up the esets.
2015-10-15 13:11:18 INFO::including GSE11237_eset
2015-10-15 13:11:18 INFO::including GSE12225.GPL3676_eset
2015-10-15 13:11:18 INFO::including GSE12945_eset
2015-10-15 13:11:19 INFO::including GSE13067_eset
2015-10-15 13:11:19 INFO::including GSE13294_eset
2015-10-15 13:11:19 INFO::including GSE14095_eset
2015-10-15 13:11:20 INFO::including GSE14333_eset
2015-10-15 13:11:20 INFO::including GSE16125.GPL5175_eset
2015-10-15 13:11:20 INFO::including GSE17536_eset
2015-10-15 13:11:20 INFO::including GSE17537_eset
2015-10-15 13:11:21 INFO::including GSE17538.GPL570_eset
2015-10-15 13:11:21 INFO::including GSE18105_eset
2015-10-15 13:11:21 INFO::including GSE2109_eset
2015-10-15 13:11:22 INFO::including GSE21510_eset

```

```

2015-10-15 13:11:22 INFO::including GSE21815_eset
2015-10-15 13:11:22 INFO::including GSE24549.GPL5175_eset
2015-10-15 13:11:22 INFO::including GSE24550.GPL5175_eset
2015-10-15 13:11:22 INFO::including GSE2630_eset
2015-10-15 13:11:22 INFO::including GSE26682.GPL570_eset
2015-10-15 13:11:22 INFO::including GSE26682.GPL96_eset
2015-10-15 13:11:22 INFO::including GSE26906_eset
2015-10-15 13:11:23 INFO::including GSE27544_eset
2015-10-15 13:11:23 INFO::including GSE28702_eset
2015-10-15 13:11:23 INFO::including GSE3294_eset
2015-10-15 13:11:23 INFO::including GSE33113_eset
2015-10-15 13:11:23 INFO::including GSE39582_eset
2015-10-15 13:11:24 INFO::including GSE3964_eset
2015-10-15 13:11:24 INFO::including GSE4045_eset
2015-10-15 13:11:24 INFO::including GSE4526_eset
2015-10-15 13:11:24 INFO::including GSE45270_eset
2015-10-15 13:11:24 INFO::including TCGA.COAD_eset
2015-10-15 13:11:24 INFO::including TCGA.READ_eset
2015-10-15 13:11:24 INFO::including TCGA.RNASeqV2.READ_eset
2015-10-15 13:11:24 INFO::including TCGA.RNASeqV2_eset
2015-10-15 13:11:25 INFO::Ids with missing data: GSE2630_eset, 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 `curatedCRCDData` (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 `curatedCRCDData` 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



```
[1] "/tmp/RtmpZMFd8j/file3ba134c2bc82"  
> 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 curatedCRCDData manual, here is a simple way to do it. For one dataset:

```
> library(curatedCRCDData)  
> 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 3.2.2 (2015-08-14), x86\_64-pc-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, stats, utils
- Other packages: Biobase 2.30.0, BiocGenerics 0.16.0, affy 1.48.0, curatedCRCDData 2.2.0, genefilter 1.52.0, logging 0.7-103, mgcv 1.8-7, nlme 3.1-122, survival 2.38-3, sva 3.18.0, xtable 1.7-4
- Loaded via a namespace (and not attached): AnnotationDbi 1.32.0, BiocInstaller 1.20.0, BiocStyle 1.8.0, DBI 0.3.1, IRanges 2.4.0, Matrix 1.2-2, RSQLite 1.0.0, S4Vectors 0.8.0, XML 3.98-1.3, affyio 1.40.0, annotate 1.48.0, grid 3.2.2, lattice 0.20-33, preprocessCore 1.32.0, splines 3.2.2, stats4 3.2.2, tools 3.2.2, zlibbioc 1.16.0

Study	PMID	N samples	curated
GSE11237	18653328	23	0
GSE12225	18959792	42	0
GSE12945	19399471	62	0
GSE13067	19088021	74	0
GSE13294	19088021	155	0
GSE14095	21680303	189	0
GSE14333	19996206	290	0
GSE16125	19672874	36	0
GSE17536	19914252	177	0
GSE17537	19914252	55	0
GSE17538	19914252	232	0
GSE18105	20162577	111	0
GSE2109	PMID unknown	427	0
GSE21510	21270110	148	0
GSE21815	21862635	141	0
GSE24549	21619627	83	0
GSE24550	21619627	90	0
GSE2630	17390049	16	0
GSE26682	21300766	156	0
GSE26906	21300766	155	0
GSE27544	22496922	90	0
GSE28702	PMID unknown	22	0
GSE3294	22095227	83	0
GSE33113	16773188	24	0
GSE39582	22496204	96	0
GSE3964	23700391	566	0
GSE4045	16542501	29	0
GSE4526	16819509	37	0
GSE45270	19016304	36	0
TCGA.COAD	PMID unknown	13	0
TCGA.READ	22810696	130	0
TCGA.RNASEqV2	22810696	51	0
TCGA.RNASEqV2	22810696	6	0
TCGA.RNASEqV2	22810696	195	0

Table 1: Datasets provided by curatedCRCDATA.