curatedBreastData Manual

Katie Planey

April 27, 2022

Contents

1 Introduction

curatedBreastData contained 34 high-quality GEO gene expression microarray datasets, all with advanced breast cancer. These datasets all contain at least one survival and/or treatment response variable, and minimal treamtent information (such as whether they had chemotherapy or not.) Clinical variables were semantically normalized across all datasets to provide a powerful database for investigating genes that are related to clinical variables such as pathological complete response, ER and HER2 IHC pathology tests, pam50 subtyping tests (when available), and tumor stage.

This database was original designed as a MySQL database, but has been re-represented as S4 ExpressionObjects with attached clinical data (phenoData) for easier data analyses in R.

1.1 Prerequisites

The package **curatedBreastData** requires minimal external packages; it requires impute.knn from the *impute* package and the *ggplot2* package for optional plots during data matrix processing, and the CRAN package XML (which uses RCurl). It also uses *Biobase*, Bioconductor's base package, for the ExpressionSet and its accessor functions.

1.2 The database

The schema of both the microarray datasets and the corresponding clinical data is detailed in proceedings from the 2013 AMIA Joint Summit on Translational Bioinformatics Planey `et al.` titled "Database integration of 4923 publicly-available samples of breast cancer molecular and clinical data.". The data presented in this package contains 2719 samples; the "missing" samples are from a dataset that is only available via a restricted data request (the 2000-sample METABRIC study) and thus is not given here (however any researcher can request the data via EBI: http://www.ebi.ac.uk/)

Whenever possible, raw data was used. Authors of linked publications to each GEO dataset were directly contacted for additional clinical information, and we are grateful for the authors who responded and kindly shared additional clinical data that we added to this package.

The database schema is shown in the figure below.

Careful processing steps were taken before any normalization, such as removing duplicated patient samples within and between datasets. For example, GSE2226, the UCSF I-SPY study, actually had samples that overlapped with samples from GSE25055, an MD-Anderson study. While rare, the author took care to keep original I-SPY samples only in the GSE22226 dataset and remove them from the GSE25055 dataset.

Additionally, samples names were inspected to infer whether clear batches within a dataset existed. Often times, a multi-site GEO dataset would have samples labeled with the patient name and the country, such as "Spain" or "Portugal". Such data tags imply the samples were collected at different sites. When such data was available, batch datasets were treated as separate datasts. GSE2226 also contained batches, but because two microarray platforms were used. These batch effects can indeed lead to differences in gene expression levels; especially for a meta-analysis, these should be treated as separate datasets.

A real-life example of this for a selected gene across several datasets in the package is shown below. The effect size for each dataset is measured. As one can see, the two different tissue collection sites for study 4 and the two differnt platforms used in study 7 manifest themselves in clear batches in both studies that result in differing microarray expression levels. Thus, they are treated as two, and not one, dataset for each of these studies.

Figure 1: Effect sizes of a selected gene showing how different sites (batches) from the same study can have different end effect size results (with survival as the meta-analysis outcome variable.)

The data objects in the curatedBreastDataExprSetList list provided in this package have already been normalized according to the protocol outlined below. Original scripts used for each original dataset can be found on the Github repository https://github.com/kplaney/curatedBreastCancer. This repository also contains results from a data quality check test in RMarkdown, and more notes on microarray processing.

2 Analysis

Gene expression datasets must be post-processed after normalization; **curatedBreastData** provides functions for removing samples and genes that have high NA rates, imputing missing values, collapsing duplicated genes or probes, removing duplicated patient samples, and filtering genes by variance. Documentation for each function is providing in its corresponding Rd documentation file. The wrapper function `processExpressionSetList()` completes all these post-processing steps on a list of S4 ExpressionSet objects like the curatedBreastDataExprSetList.rda list provided in this package.

This package works with expression data in the form of S4 ExpressionSet objects and/or lists of S4 ExpressionSet objects. In this example workflow, we load up all of the datasets.

```
> library("curatedBreastData")
> #load up datasets that are in S4 expressionSet format.
> #clinical data from master clinicalTable already linked to each sample
> #in these ExpressionSets in the phenoData slot.
> data(curatedBreastDataExprSetList);
```

We can see that clinical data for each dataset is included in the phenoData slot.

```
> #check out the clinical data for dataset 3
> #first look at the GEO study name
> names(curatedBreastDataExprSetList)[3]

[1] "study_4913_GPL3558_all"

> #only take the first 3 patients for sake of printing to screen
> #look at first 10 clinical variables
> head(pData(curatedBreastDataExprSetList[[3]])[c(1:3), c(1:10)])

                  datasetName dbUniquePatientID study_ID.x patient_ID GEO_GSMID
110388 study_4913_GPL3558_all               597       4913     110388    110388
110392 study_4913_GPL3558_all               598       4913     110392    110392
110394 study_4913_GPL3558_all               599       4913     110394    110394
```

```
       platform_ID GEO_platform_ID AE_platform_ID coordinating_GSE_series_GSMID
110388        3558         GPL3558           <NA>                            NA
110392        3558         GPL3558           <NA>                            NA
110394        3558         GPL3558           <NA>                            NA
       original_study_ID
110388         wsb 10167
110392          wsb 1281
110394          wsb 1319
```

We then post-process this data. Let's say we only want the top 5000 genes by variance for each dataset. This can take a few minutes as we're post-processing 34 datasets. This takes a while, so it's not included in the examples() section.

```
> #process only the first two datasets to avoid a long-running example:
> #take top 5000 genes by variance from each dataset.
> proc_curatedBreastDataExprSetList <- processExpressionSetList(exprSetList=curatedBreastDataExprSetList[1:2
```

Now we have processed expression matrices, each with the top 5000 genes by variance. We can then begin to explore what clinical variables we could use to run a meta-analysis or supervised analysis with all of this data.

All clinical variables are semantically normalized/have the same names across all datasets in the list. This is helpful, but oftentimes, we want to know if we have enough patients with certain clinical variables across all datasets. The global clinicalTable data frame provided in curatedBreastData combines all of the clinical data across all datasets.

Treatment data has been carefully curated and aggregtaed into semantically normalized variables so that a user can both determine which patients received a therapy, and then control for these variables in analyses. For example, using the master clinicalData table, we can quickly determine how many patients ever had chemotherapy. The user can also inspect specific classes of chemotherapy, such as taxanes. Whenever granular therapy information was provided, it was recorded in the clinicalTable (otherwise, it was left as NA.)

```
> #load up master clinical data table
> data(clinicalData)
> #look at some of the clinical variable name definitions
> clinicalData$clinicalVarDef[c(1:2),]

                      variableName
dbUniquePatientID dbUniquePatientID
study_ID                  study_ID
                             definition.NA.means.not.recorded
dbUniquePatientID Unique patient id created for this database.
study_ID                                       GEO GSE study ID.

> #Check out the treatment information.
> #just do first three patients
> head(clinicalData$clinicalTable)[c(1:3),
+                                  c(112:ncol(clinicalData$clinicalTable))]

  aromatase_inhibitor estrogen_receptor_blocker
2                   0                         0
3                   0                         0
4                   0                         0
```

```
  estrogen_receptor_blocker_and_stops_production
2                                              0
3                                              0
4                                              0
  estrogen_receptor_blocker_and_eliminator anti_HER2 tamoxifen doxorubicin
2                                        0         0         0           0
3                                        0         0         0           0
4                                        0         0         0           0
  epirubicin docetaxel capecitabine fluorouracil paclitaxel cyclophosphamide
2          1         0            0            1          1                1
3          1         0            0            1          1                1
4          1         0            0            1          1                1
  anastrozole fulvestrant gefitinib trastuzumab letrozole chemotherapy
2           0           0         0           0         0            0
3           0           0         0           0         0            0
4           0           0         0           0         0            0
  hormone_therapy no_treatment methotrexate cetuximab carboplatin other
2               0            0            0         0           0     0
3               0            0            0         0           0     0
4               0            0            0         0           0     0
  taxaneGeneral neoadjuvant_or_adjuvant study_specific_protocol_number
2             0                     neo                              1
3             0                     neo                              1
4             0                     neo                              1
```

```
> #how many had chemotherapy?
> numChemoPatients <- length(which(
+   clinicalData$clinicalTable$chemotherapyClass==1))
> #around 1500 had chemotherapy
> numChemoPatients
```

```
[1] 1495
```

```
> #which patients specifically had a taxane chemotherapy?
> numChemoTaxane <- length(which(clinicalData$clinicalTable$taxane==1))
> numChemoTaxane
```

```
[1] 1218
```

We can also look at adjuvant vs. neoadjuvant therapy.

```
> #how many had adjuvant therapy?
> numAdjPatients <- length(which(
+   clinicalData$clinicalTable$neoadjuvant_or_adjuvant=="adj"))
> #over a 1000 had (documented) adjuvant therapy
> numAdjPatients
```

```
[1] 1136
```

The clinicalTable data frame can allow us to then select the relevant patients from each dataset who have the outcomes variable we want. For example, we inspect the number of samples with data pertaining to Overall Survival below.

```
> #how many patients have non-NA OS binary data?
> length(which(!is.na(clinicalData$clinicalTable$OS)))

[1] 409

> #how many have OS data in the more granular form of months until OS?
> #this variable includes studies that had a cieling for tracking OS
> length(which(!is.na(clinicalData$clinicalTable$OS_months_or_MIN_months_of_OS)))

[1] 406

> #how many patients have OS information that is definitively
> #followed up until their death (details on how studies collect OS data can be surprising!)
> length(which(!is.na(clinicalData$clinicalTable$OS_up_until_death)))

[1] 211

> #finish up with sessionInfo
> sessionInfo()

R version 4.2.0 RC (2022-04-21 r82226)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 20.04.4 LTS

Matrix products: default
BLAS:   /home/biocbuild/bbs-3.16-bioc/R/lib/libRblas.so
LAPACK: /home/biocbuild/bbs-3.16-bioc/R/lib/libRlapack.so

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_GB              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] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] curatedBreastData_2.25.0 BiocStyle_2.25.0         Biobase_2.57.0
[4] BiocGenerics_0.43.0      impute_1.71.0            ggplot2_3.3.5
[7] XML_3.99-0.9

loaded via a namespace (and not attached):
 [1] pillar_1.7.0        compiler_4.2.0      BiocManager_1.30.17
 [4] tools_4.2.0         digest_0.6.29       evaluate_0.15
 [7] lifecycle_1.0.1     tibble_3.1.6        gtable_0.3.0
[10] pkgconfig_2.0.3     rlang_1.0.2         cli_3.3.0
[13] DBI_1.1.2           yaml_2.3.5          xfun_0.30
[16] fastmap_1.1.0       withr_2.5.0         dplyr_1.0.8
[19] knitr_1.38          generics_0.1.2      vctrs_0.4.1
[22] grid_4.2.0          tidyselect_1.1.2    glue_1.6.2
[25] R6_2.5.1            fansi_1.0.3         rmarkdown_2.14
```

```
[28] purrr_0.3.4         magrittr_2.0.3      scales_1.2.0
[31] ellipsis_0.3.2      htmltools_0.5.2     assertthat_0.2.1
[34] colorspace_2.0-3    utf8_1.2.2          munsell_0.5.0
[37] crayon_1.5.1
```

3 References

1. Planey, Butte. Database integration of 4923 publicly-available samples of breast cancer molecular and clinical data. AMIA Joint Summits Translational Science Proceedings. (2003) PMC3814460
2. Github repo with code, further documentation on datasets and baseline normalization schemes, and database quality checks: https://github.com/kplaney/curatedBreastCancer