hgu133plus2CellScore 1.30.0: Standard ExpressionSet for CellScore [hgu133plus2]

Nancy Mah, Katerina Taškova
nancy.l.mah@googlemail.com, katerina@tashkova.org

November 4, 2025

Contents

1 Introduction

The **hgu133plus2CellScore** package contains a dataset of manually curated transcription profiles [1] from the Affymetrix Human Genome U133 Plus 2.0) microarray platform. This dataset contains expression data from normal tissues or cell types, and is used as a reference dataset for evaluation of cell identity using the **CellScore** package.

2 Installation

This vignette assumes that you have already installed R ($\geq$ 4.5.1) and that you have basic working knowledge of R [2]. You will additionally need to install the core Bioconductor packages if these have not already been installed:

```
if (!requireNamespace("BiocManager", quietly=TRUE))
    install.packages("BiocManager")
BiocManager::install()
```

To install the **hgu133plus2CellScore** package from the Bioconductor repository:

```
BiocManager::install("hgu133plus2CellScore")
```

Load the packages and standard dataset:

```
library(Biobase)
library(hgu133plus2CellScore)
```

The hgu133plus2CellScore dataset is stored in an ExpressionSet object:

```
eset.std

## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 19851 features, 837 samples
##   element names: calls, exprs
## protocolData: none
## phenoData
##   sampleNames: GPL570.GSE11350.GSM282008
##     GPL570.GSE11350.GSM282009 ... GPL570.GSE62962.GSM1537250
##     (837 total)
##   varLabels: experiment_id sample_id ...
##     donor_cell_body_location (11 total)
##   varMetadata: labelDescription
## featureData
##   featureNames: 1 2 ... 101978719 (19851 total)
##   fvarLabels: probe_id median ... entrezgene_id (6 total)
##   fvarMetadata: labelDescription
## experimentData: use 'experimentData(object)'
## Annotation:
```

For detailed examples demonstrating the usage of the **hgu133plus2CellScore** package, please see the tutorial for **CellScore**.

3 How the hgu133plus2CellScore data package was assembled

Although the process below is described for Affymetrix 3'IVT microarrays, the general procedure could be modified to accomodate other data from other microarray platforms or RNA-seq datasets. We currently do not recommend combining standard datasets from multiple platforms as this has not been extensively tested for use with the **CellScore** package.

Data wrangling of the standard dataset includes the following steps:

1. *Select suitable samples from one expression platform*
2. *Manually annotate the samples*
3. *Process the raw microarray data*
4. *Microarray data quality control*
5. *Associate gene annotations to probe IDs*
6. *Create the ExpressionSet object*

The following description assumes that the user has some familiarity with `R` and expression data analysis in Bioconductor. The vignette is not intended to be a comprehensive tutorial.

3.1 Select suitable samples from one expression platform

Data from a wide variety of cell types or tissues can be searched in public databases such as Gene Expression Omnibus(GEO) or Array Express. Preferably, the samples should be as "normal" as possible. For example, neither samples with genetic modifications such as reporter gene construct inserts, nor samples from individuals with disease should be considered as standard samples. Standards are usually tissue samples or cell lines; however, carefully selected engineered or derived cell types could also be used as standard samples if you want to use these cell types as a basis for comparison.

3.2 Manually annotate the samples

A *phenotype data frame*, named here as `phenotype.data.frame`, should be created with information about the samples in the expression matrix. Samples are in rows, and columns are attributes of the sample. The row names of the data frame must be unique sample IDs and must exactly match the column names of the `callsSub.matrix` and `normalizedSub.matrix` (see section 3.5). The *phenotype data frame* must contain the following columns:

- `experiment_id`: This should be a unique identifier for an experiment, for example a GEO experiment ID or an ArrayExpress experiment ID.
- `sample_id`: Sample IDs should be unique, such as the GSM accession numbers from GEO.
- `platform_id`: Use a unique ID for each platform, such as GPL accession numbers from GEO.

- `cell_type`: Use a short text description here.
- `category`: Each sample can be assigned to one of "standard"/"test"/"NA". For the purposes of the standard dataset, all samples should be "standard".
- `general_cell_type`: Use an abbreviation to label the general cell type. Preferably there should be no spaces or punctuation in the abbreviation. For example, FIB for fibroblast.
- `donor_tissue`: If the sample is a derived cell type, then enter the abbreviation for the donor cell type. If the sample is standard cell type, then enter the donor tissue from which it was isolated. Otherwise, enter "NA".
- `sub_cell_type1`: The sub-cell type is a compound term of the general cell type and its donor tissue.

Additional columns are optional. The properties in these columns could be used to color the individual samples in the rug plots generated by the **CellScore** functions. For example, the samples could be colored by

- `transition_induction_method`, the method used to engineer the derived cell types, or
- `donor_cell_body_location`, the anatomical area from which the donor cell was taken.

3.3 Process the raw microarray data

The raw data (*.CEL files) should be first background corrected by **affy** [3] and normalized using the **YuGene** transform [4], which enables the addition of more samples without having to pre-process the whole dataset every time a new sample is added. Of course, you are free to use any other normalization method as you like (e.g. RMA [5] or fRMA [6] combined with ComBat [7]); it is just important for consistency that all the standard samples (and test samples) are processed in the same manner, and that some form of batch correction is applied. Next, the detection p-values were calculated using the function `affy::mas5calls()`. In the present data package, a probeset was considered "Present" if the detection p-value was less than 0.05, and "Absent" if the detection p-value was greater than or equal to 0.05.

Here are some example commands on how to process the *.CEL files:

```
## Install affy
if (!requireNamespace("BiocManager", quietly=TRUE))
    install.packages("BiocManager")
BiocManager::install("affy")
## Install YuGene
install.packages("YuGene")

## Load the packages
library(affy)
library(YuGene)
```

```
## Read in all *.CEL files in the current working directory;
## if the files are not in the working directory then uncomment
## the first command line and then read the files. Note that
## path_to_CEL should be replaced with path to the directory
## containing the *.CEL files

#setwd(path_to_CEL)
data <- ReadAffy()

## Background noise correction
bg.corr <- expresso(data, bg.correct=TRUE, bgcorrect.method="rma",
                    normalize=FALSE, pmcorrect.method="pmonly",
                    summary.method="avgdiff")
## Log2-transform the background corrected data
bg.corr.log2 <- log2(bg.corr)
## Perform YuGene transform (that results with
## normalized expression values between 0 and 1)
normalized.matrix <- YuGene(bg.corr.log2)

## Calculate mas5 present/absent calls
co <- mas5calls(data)
co <- assayData(co)[["se.exprs"]] #extract detection p-values
pvalue.detection.cutoff <- 0.05
calls.matrix <- co < pvalue.detection.cutoff
```

The YuGene-transformed expression matrix (`normalized.matrix`) and the binary matrix of the present/absent calls (`calls.matrix`) should have the same dimensions, and all row and columns should be in exactly the same order. The rownames of each matrix should be probeset IDs.

3.4 Microarray data quality control

Despite careful manual curation of the samples, there can still be outliers that are unsuitable to serve as standard samples. For example, a sample may be too degraded or wrongly annotated. Diagnostic boxplots, density plots, and principal component analysis plots of the raw and/or normalized data may help here to identify samples which should be eliminated as standard samples. We refer the user to the packages **affy** and **arrayQualityMetrics** [8] on how to generate these diagnostic plots.

3.5 Associate gene annotations to probe IDs

Gene annotation may come from the microarray manufacturer or from annotation data packages in Bioconductor (e.g. NCBI (**org.Hs.eg.db** [9]) or Ensembl (**biomaRt** [10]). To annotate the present data package, we used the Annotation Data package **org.Hs.eg.db**. For simplicity, we restricted the final standard expression and calls dataset to one Affymetrix probeset per Entrez Gene ID, in order to generate a non-redundant gene table. In the case that multiple Affymetrix probesets mapped to the same gene, the probeset with the highest median expression across all samples was chosen to represent that gene. The non-redundant subsets of expression and calls matrices (named here as `normalizedSub.matrix` and `callsSub.matrix`, respectively), along with the

corresponding annotaion data frame (named here as `annotation.data.frame`), must have the same number of rows and the rows must be in the same order.

Whatever your source of annotation, it should be organized in a data frame (`annotation.data.frame`) with the following columns:

- `probe_id`: this can be any unique row identifier. The row order must be the same as the *normalizedSub.matrix* and *callsSub.matrix*.
- `platform_id`: unique identifier for a microarray platform, such as the GPL accession number from Gene Expression Omnibus.
- `gene_symbol`: official gene symbol
- `gene_name`: name of the gene
- `entrezgene_id`: an integer number used as the Entrez Gene ID from NCBI

Other columns will be ignored.

3.6 Create the ExpressionSet object

Finally, all the data was assembled as an ExpressionSet object:

```
## Example code:
## Create a new assayData object from the normalized expression data
## and the calls matrix
assay.data <- assayDataNew(exprs=as.matrix(normalizedSub.matrix),
                           calls=as.matrix(callsSub.matrix) )
## Create an AnnotatedDataFrame object from the phenotype data frame
pheno.table <- new("AnnotatedDataFrame", data=phenotype.data.frame)
## Create an AnnotatedDataFrame object from the annotation data frame
annotation.table <- new("AnnotatedDataFrame", data=annotation.data.frame)
## Create the new ExpressionSet object with all the data in one object
eset.std <- ExpressionSet(assayData=assay.data,
                          phenoData=pheno.table,
                          featureData=annotation.table)
```

DONE!

4 R session information

```
sessionInfo()

## R version 4.5.1 Patched (2025-08-23 r88802)
## Platform: x86_64-pc-linux-gnu
## Running under: Ubuntu 24.04.3 LTS
##
## Matrix products: default
## BLAS:   /home/biocbuild/bbs-3.22-bioc/R/lib/libRblas.so
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.12.0  LAPACK version 3.12.0
##
## 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
##
## time zone: America/New_York
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods
## [7] base
##
## other attached packages:
## [1] hgu133plus2CellScore_1.30.0 knitr_1.50
## [3] Biobase_2.70.0              BiocGenerics_0.56.0
## [5] generics_0.1.4
##
## loaded via a namespace (and not attached):
## [1] compiler_4.5.1 tools_4.5.1    highr_0.11     xfun_0.54
## [5] evaluate_1.0.5
```

References

[1] N. Mah, K. E. Amrani, W. Zhang, H. Stachelscheid, and A. Kurtz, "Cellfinder's molecular database and its application to stem cell research," *Genomics and Computational Biology*, vol. 3, no. 1, p. e48, 2017.

[2] R. D. C. Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2007. ISBN 3-900051-07-0.

[3] L. Gautier, L. Cope, B. M. Bolstad, and R. A. Irizarry, "affy—analysis of affymetrix genechip data at the probe level," *Bioinformatics*, vol. 20, no. 3, pp. 307–315, 2004.

[4] K.-A. Le Cao, F. Rohart, L. McHugh, O. Korn, and C. A. Wells, *YuGene: A Simple Approach to Scale Gene Expression Data Derived from Different Platforms for Integrated Analyses*, 2015. R package version 1.1.5.

[5] R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed, "Exploration, normalization, and summaries of high density oligonucleotide array probe level data," *Biostatistics*, vol. 4, no. 2, pp. 249–264, 2003.

[6] M. N. McCall, B. M. Bolstad, and R. A. Irizarry, "Frozen robust multiarray analysis (frma)," *Biostatistics*, vol. 11, no. 2, pp. 242–253, 2010.

[7] W. E. Johnson, C. Li, and A. Rabinovic, "Adjusting batch effects in microarray expression data using empirical bayes methods," *Biostatistics*, vol. 8, no. 1, pp. 118–127, 2007.

[8] A. Kauffmann, R. Gentleman, and W. Huber, "arrayqualitymetrics—a bioconductor package for quality assessment of microarray data," *Bioinformatics*, vol. 25, no. 3, pp. 415–416, 2009.

[9] M. Carlson, *org.Hs.eg.db: Genome wide annotation for Human*, 2016. R package version 3.4.0.

[10] S. Durinck, Y. Moreau, A. Kasprzyk, S. Davis, B. De Moor, A. Brazma, and W. Huber, "Biomart and bioconductor: a powerful link between biological databases and microarray data analysis," *Bioinformatics*, vol. 21, pp. 3439–3440, 2005.