Feeding the output of a flow cytometry assay into cellHTS2

Florian Hahne

April 30, 2018

The package *prada* can be used to analyze flow cytometry raw data derived from cell-based assays. The output of these analyses are highly processed data or even scored hit lists. However, for some applications it might also be useful to integrate this output into the *cellHTS2* package in order to make use of its excellent visualization and QA features. Although *cellHTS2* is more geared towards the analysis of unprocessed raw data data the process is rather straight forward. To exemplify the procedure we added some sample files derived from an apoptosis assay to this package which contain all the necessary information to be provided for *cellHTS2*. The generation of these files can be accomplished using the available file handling functions provided by R or by using text processing software. The data consist of scored effect sizes (odds ratios) for two replicates of two 96 well plates. Cells in each well were transfected with a different overexpression construct for a protein of unknown function and the induction of apoptosis was measured using FACS readout. The file *Platelist.txt* maps the contents of the data files for each plate to plate and replicate identifiers. We first load the package.

```
> library("cellHTS2")
```

By calling `readPlateData` we can import the data and generate a *cellHTS* object. In the import function we also want to calculate the negative log transformation of the odds ratio to ensure symmetry around zero.

```
> experimentName = "ApoptosisScreen"
> dataPath = system.file("extdata", package = "prada")
> x = readPlateList("Platelist.txt", name = experimentName,
+                   path = dataPath, verbose = FALSE,
+                   importFun=function(file, path){
+                       data <- read.delim(file, header=FALSE, as.is=TRUE)
+                       return(list(data.frame(well=I(as.character(data[,2])), val=-log10(dat
+                                   readLines(file)))
+                   })
> x

cellHTS (storageMode: lockedEnvironment)
assayData: 192 features, 2 samples
  element names: Channel 1
phenoData
  sampleNames: 1 2
```

```
  varLabels: replicate assay
  varMetadata: labelDescription channel
featureData
  featureNames: 1 2 ... 192 (192 total)
  fvarLabels: plate well controlStatus
  fvarMetadata: labelDescription
experimentData: use 'experimentData(object)'
state:  configured = FALSE
        normalized = FALSE
        scored = FALSE
        annotated = FALSE
Number of plates: 2
Plate dimension: nrow = 8, ncol = 12
Number of batches: 1
```

In a second step we tell *cellHTS2* where to expect controls on the plates and also give some details about the experiment. This information is provided by the files *Plateconf.txt*, *Screenlog.txt* and *Description.txt*.

```
> x = configure(x, confFile="Plateconf.txt", descripFile="Description.txt", path=dataPath)
```

In the next step we include annotation information for both plates (provided by the file *GeneIDs*.

```
> geneIDFile = file.path(dataPath, "GeneIDs.txt")
> x = annotate(x, geneIDFile)
```

We ask *cellHTS2* to do a simple mean normalization even though rough normalization has already been done during our analysis. We also sumarize replicates and score.

```
> xn <- normalizePlates(x, method="mean")
> xsc <- scoreReplicates(xn, sign="-", method="zscore")
> xsc <- summarizeReplicates(xsc, summary="mean")
```

In the final step we generate the HTML report.

```
> od <- tempfile()
> writeReport(raw=x, normalized=xn, scored=xsc, force = TRUE, outdir=od)
```

The final report can now be inspected in the subfolder *ApoptosisScreen* of the temporary working directory. For more information on each individual step and the content of the report please consult the vignette of the *cellHTS2* package.