An Introduction to SingleCellAssay

Andrew McDavid

September 10, 2012

1 Philosophy

SingleCellAssayis an R/Bioconductor package for Fluidigm and friends. We seek to support assays that have multiple *features* (genes, markers, etc) per *well* (cell, etc) in a flexible format. The assays is mostly *complete* in the sense that most wells contain measurements for all features. We test for completeness, and complete the object if it is not, so very incomplete assays just make things a bit slower.

Internally, we store everything as one giant `data.frame` with names of special columns kept in a `mapping` that contains column names and keywords. It is in long-melted format, in feature-major order, so not especially fast or space-efficient, but rather is intended to be very flexible.

Each well, feature TODO: , and unit (phenotype) has covariates measured. These are kept in `AnnotatedDataframes`, which are generated from the basal `data.frame`, if so provided. TODO: If not provided, then they can be added after object creation.

2 Reading Data

Data imported in a Fluidigm instrument-specific format (whose details are undocumented, and probably subject-to-change) or in “long” (melted) format, in which each row is a measurement, so if there are N wells and M cells, then the `data.frame` should contain $N \times M$ rows.

For example, the following data set was provided in as a comma-separated value file. It has the cycle threshold (ct) recorded, with non-detected genes recorded as NAs. For the Fluidigm/qPCR single cell expression functions to work as expected, we must report the expression threshold ($c_{\max} - ct$), which is proportional to the log-expression.

```r
library(SingleCellAssay)

## Scalable Robust Estimators with High Breakdown Point (version 1.3-02)

data(vbeta)
vbeta <- within.data.frame(vbeta, {
    Et <- 40 - Ct
    Et <- ifelse(is.na(Et), 0, Et)
})
vbeta.fa <- FluidigmAssay(vbeta, idvars = c("Subject.ID", "Chip.Number", "Well"),
    primerid = "Gene", measurement = "Et", ncells = "Number.of.Cells", geneid = "Gene",
    cellvars = c("Number.of.Cells", "Population"), phenovars = c("Stim.Condition",
        "Time"), id = "vbeta all")
show(vbeta.fa)

## FluidigmAssay  id:  vbeta all
##  456  wells;  75  features
```

We specify `vbeta`, as the `data.frame` from which the `FluidigmAssay` object will be created, the `idvars` which is a column(s) in `vbeta` that unique identify a well, the `primerid`, which is a column(s) that specify which feature is measured at this nrow. The `measurement` gives the column name containing the log-expression measurement, `ncells` contains the number of cells (or other normalizing factor) for the well. `geneid`, `cellvars`, `phenovars` all specify additional columns to be included in the `featureData`, `phenoData` and `cellData` (TODO: wellData):

```
head(fData(vbeta.fa))

##          Gene
## B3GAT1 B3GAT1
## BAX       BAX
## BCL2     BCL2
## CCL2     CCL2
## CCL3     CCL3
## CCL4     CCL4

head(cData(vbeta.fa))

##   Number.of.Cells            Population Subject.ID Chip.Number Well
## 1               1 CD154+VbetaResponsive      Sub01           1  A01
## 2               1 CD154+VbetaResponsive      Sub01           1  A02
## 3               1 CD154+VbetaResponsive      Sub01           1  A03
## 4               1 CD154+VbetaResponsive      Sub01           1  A04
## 5               1 CD154+VbetaResponsive      Sub01           1  A05
## 6               1 CD154+VbetaResponsive      Sub01           1  A06
##   Stim.Condition Time
## 1      Stim(SEB)   12
## 2      Stim(SEB)   12
## 3      Stim(SEB)   12
## 4      Stim(SEB)   12
## 5      Stim(SEB)   12
## 6      Stim(SEB)   12
```

3 Subsetting, splitting, combining

It's possible to subset `SingleCellAssay`objects by wells TODO: and features. Double square brackets ("[[") and `subset` subset by wells. Both integer and boolean indices may be used. The usual recycling rules (if the index is shorter than the number of rows) apply. TODO: Single square brackets subset by [wells, features].

```
sub1 <- vbeta.fa[[1:10]]
show(sub1)

## FluidigmAssay  id:  vbeta all
##  10  wells;  75  features

sub2 <- subset(vbeta.fa, Well == "A01")
show(sub2)

## FluidigmAssay  id:  vbeta all
##  5  wells;  75  features
```

A `SingleCellAssay`may be split into a list of `SingleCellAssay`, which is known as a `SCASet`.

```
sp1 <- split(vbeta.fa, "Subject.ID")

## Warning:  namedlist should not be empty
## Warning:  namedlist should not be empty

show(sp1)

## SCASet of size  2
## Samples  Sub01, Sub02
```

```
sp2 <- split(vbeta.fa, factor(rbinom(nrow(vbeta.fa), 1, prob = 0.2)))

## Warning:  namedlist should not be empty
## Warning:  namedlist should not be empty

show(sp2)

## SCASet of size  2
## Samples  0, 1
```

The splitting variable can either be a character vector naming column(s) of the `SingleCellAssay`, or may be a `factor` or `list` of `factors`.

It's possible to combine `SingleCellAssay`.