How to apply the ddCt method

Markus Ruschhaupt

January 29, 2007

In this vignette we describe how the functions `ddCt`, `readSDM` and `eSetWrite`, available in the package *prada*, can be used to perform the ddCt method for a given set of data. First the files including the data must be specified. Here the files 'Experiment1.txt' and 'Experiment2.txt'are the output files of SDS, Version 2.1. Furthermore it is possible to include sample specific information. In this example these informations are stored as the tab-delimited text file 'sampleData.txt'. The warnings that appear during the calculation process, e.g. if the Ct value for the housekeeping gene is missing, are saved as a text file specified by the parameter 'warningFile'.

```
> library(prada)
> datadir <- function(x) file.path(.path.package("prada"), "extdata",
+     x)
> savedir <- function(x) file.path(.path.package("prada"), "extdata",
+     x)
> file.names <- c(datadir("Experiment1.txt"), datadir("Experiment2.txt"))
> info <- datadir("sampleData.txt")
> warningFile <- savedir("warning.output.txt")
```

The housekeeping gene and the calibration sample/s must be specified. If there is more than one calibration sample the mean of the Ct values will be uses for calibration.

```
> name.referenz.sample <- c("Sample1")
> name.referenz.gene <- "Gene2"
```

Now the data from the experiment files will be read.

```
> library(Biobase)
> CtData <- readSDM(file.names)
> sampleInformation <- read.phenoData(info, header = TRUE)
```

With this information we can start the calculation of the ddCt method.

```
> result <- ddCt(CtData, calibrationSample = name.referenz.sample,
+     housekeepingGene = name.referenz.gene, sampleInformation = sampleInformation,
+     filename = warningFile)
```

`result` is a member of the class *eSet* from the *Bioconductor* package. The function `ddCt` performs several steps to calculate the ddCt and other values.

1. The median of each triplet is the Ct value.
2. The Ct values of the housekeeping gene are subtracted from the other corresponding Ct values. The results are the dCt values.
3. The dCt values of the calibration sample (or the mean of the dCt values of several calibration samples) are subtracted from the other corresponding dCt values. This gives the ddCt values.
4. The transformation $x \to 2^{-x}$ is applied to the ddCt values. This gives the levels.

Additionally the MAD of each triplet is calculated (if you have three different points the MAD is the shorter of the two neighbour points distances). The longer distance(if there is a triplet without NA values) and the number of NA values for each triplet is also calculated. All these values are stored in the corresponding slots of the eList of the `result` object. For example the MAD of all triplets is accessible through

```
> assayData(result)$Ct.error
```

```
        Sample
Detector    Sample1   Sample2    Sample3    Sample4
   Gene1 0.44017062 0.6452858 0.17610413 0.11620050
   Gene2 0.06299529 0.0954406 0.03288257 0.03057597
   Gene3 0.72107613 0.1657458 0.45246203 0.05064879
```

We save the results as tab-delimited text files. Therefor we use the function `eListWrite`.

```
> eListWrite(result, file = savedir("allValues.txt"), sep = "\t",
+     row.names = FALSE)
> write.table(exprs(result), file = savedir("level_matrix.txt"),
+     sep = "\t", col.names = NA)
> write.table((result)$dCt, file = savedir("dCt_matrix.txt"), sep = "\t",
+     col.names = NA)
```