ReadqPCR: Functions to load RT-qPCR data into R

James Perkins
University College London

Matthias Kohl
Furtwangen University

March 23, 2013

Contents

1 Introduction 1
2 read.LC480 2
3 read.qPCR 7
4 read.taqman 8
5 qPCRBatch 9

1 Introduction

The package "ReadqPCR" contains different functions for reading qPCR data into R.

As well as the functions to read in the data, "ReadqPCR" contains the qPCRBatch and CyclesSet class definition. The data output by these RT-qPCR systems is in the form of fluorescence values or quantification cycle (Cq values), which represents the number of cycles of amplification needed in order to detect the expression of a given gene from a sample.

"ReadqPCR" is designed to be complementary to another R package, "NormqPCR", which is intended for the normalisation of qPCR data. It must be installed before the other module.

We load the "ReadqPCR" package.

> library(ReadqPCR)
2 read.LC480

With function read.LC480 raw fluorescence data of Roche LightCycler 480 may be read in. The result is saved in an object of class CyclesSet. At the moment the function works only with the data exported to a txt-file, but we plan to add support also for the xml-export.

```r
> path <- system.file("exData", package = "ReadqPCR")
> LC480.example <- file.path(path, "LC480_Example.txt")
> cycData <- read.LC480(file = LC480.example)
> ## Fluorescence data
> head(exprs(cycData))

   B1   B2   B3   B4   B5   B6   B7   B8   B9  B10  B11 C1  C2  C3  C4  C5  C6  C7  C8  C9
1 16.61 17.74 17.57 17.67 29.30 41322 41625.00 26.84 16.93 16.64 16.49 16.55 27.43 18.14 16.77 27.19 17.50 41442.00 28.19
2 16.67 17.80 17.42 17.61 17.71 29.76 41322 17.19 27.58 41625.00 16.68 16.57 17.78 27.47 17.64 17.73 29.84 41322 17.16 27.58 41350.00 16.64
3 16.56 17.72 27.49 17.57 17.64 29.87 41442 17.22 27.61 17.14 16.57 16.52 17.68 27.44 17.59 17.62 29.78 17 41625.00 27.56 41472.00 16.57
4 16.59 17.69 27.50 17.54 17.62 29.90 41564 17.22 27.61 17.14 16.56 16.52 17.68 27.44 17.59 17.62 29.78 17 41625.00 27.56 41472.00 16.57
5 16.59 17.69 27.50 17.54 17.62 29.90 41564 17.22 27.61 17.14 16.56 16.59 17.69 27.50 17.54 17.62 29.90 41564 17.22 27.61 17.14 16.56
6 16.59 17.69 27.50 17.54 17.62 29.90 41564 17.22 27.61 17.14 16.56 26.35 16.89 16.55 27.43 18.14 16.77 27.19 17.50 41442.00 28.19
7 26.35 16.89 16.55 27.43 18.14 16.77 27.19 17.50 41442.00 28.19 26.21 16.86 16.54 27.46 41473.00 16.81 41574.00 17.50 16.99 28.22
8 26.27 16.81 16.51 27.38 41504.00 16.83 27.17 17.43 41564.00 41606.00 26.22 16.79 16.51 27.43 41443.00 16.87 27.16 17.47 41322.00 28.15
9 26.22 16.79 16.51 27.43 41443.00 16.87 27.16 17.47 41322.00 28.15 26.15 16.84 16.44 27.47 41462.00 16.86 27.14 17.51 41291.00 28.17
10 26.15 16.84 16.44 27.47 41462.00 16.86 27.14 17.51 41291.00 28.17 21 141443.00 16.49 25.92 17.22 16.80 27.25 17.66 17.18 27.41 17.42 17.22
11 141443.00 16.49 25.92 17.22 16.80 27.25 17.66 17.18 27.41 17.42 17.22 26.35 16.89 16.55 27.43 18.14 16.77 27.19 17.50 41442.00 28.19
12 26.35 16.89 16.55 27.43 18.14 16.77 27.19 17.50 41442.00 28.19 26.21 16.86 16.54 27.46 41473.00 16.81 41574.00 17.50 16.99 28.22
13 26.27 16.81 16.51 27.38 41504.00 16.83 27.17 17.43 41564.00 41606.00 26.22 16.79 16.51 27.43 41443.00 16.87 27.16 17.47 41322.00 28.15
14 26.22 16.79 16.51 27.43 41443.00 16.87 27.16 17.47 41322.00 28.15 26.15 16.84 16.44 27.47 41462.00 16.86 27.14 17.51 41291.00 28.17
15 26.15 16.84 16.44 27.47 41462.00 16.86 27.14 17.51 41291.00 28.17 21 141443.00 16.49 25.92 17.22 16.80 27.25 17.66 17.18 27.41 17.42 17.22
16 141443.00 16.49 25.92 17.22 16.80 27.25 17.66 17.18 27.41 17.42 17.22 26.35 16.89 16.55 27.43 18.14 16.77 27.19 17.50 41442.00 28.19
17 26.35 16.89 16.55 27.43 18.14 16.77 27.19 17.50 41442.00 28.19 26.21 16.86 16.54 27.46 41473.00 16.81 41574.00 17.50 16.99 28.22
18 26.27 16.81 16.51 27.38 41504.00 16.83 27.17 17.43 41564.00 41606.00 26.22 16.79 16.51 27.43 41443.00 16.87 27.16 17.47 41322.00 28.15
19 26.22 16.79 16.51 27.43 41443.00 16.87 27.16 17.47 41322.00 28.15 26.15 16.84 16.44 27.47 41462.00 16.86 27.14 17.51 41291.00 28.17
20 26.15 16.84 16.44 27.47 41462.00 16.86 27.14 17.51 41291.00 28.17
```
```
> ## Pheno data
> head(pData(cycData))

<table>
<thead>
<tr>
<th>Sample position</th>
<th>Sample name</th>
<th>Program number</th>
<th>Segment number</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>A1 Sample_1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>A2</td>
<td>A2 Sample_2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>A3</td>
<td>A3 Sample_2</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

> # Pheno data
> head(pData(cycData))
The class CyclesSet is an extension of class eSet.

> cycData

CyclesSet (storageMode: lockedEnvironment)
assayData: 45 features, 96 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: A1 A2 ... H12 (96 total)
  varLabels: Sample position Sample name Program number Segment number
  varMetadata: labelDescription
featureData
  featureNames: 1 2 ... 45 (45 total)
  fvarLabels: Cycle number Acquisition time Acquisition temperature
  fvarMetadata: labelDescription
experimentData: use 'experimentData(object)'
Annotation:

> getClass("CyclesSet")

Class "CyclesSet" [package "ReadqPCR"]

Slots:

Name: assayData phenoData featureData
The additional sample information file can be read in with function `read.LC480SampleInfo` and is saved in an object of class `AnnotatedDataFrame`.

```r
> LC480.SamInfo <- file.path(path, "LC480_Example_SampleInfo.txt")
> samInfo <- read.LC480SampleInfo(file = LC480.SamInfo)
> ## Additional sample information
> head(pData(samInfo))
```

<table>
<thead>
<tr>
<th>Sample position</th>
<th>Sample name</th>
<th>Replicate of</th>
<th>Filter combination</th>
<th>Target name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1</td>
<td>465-510</td>
<td>negativ Kontrolle</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>A2</td>
<td>465-510</td>
<td>goi</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>A3</td>
<td>465-510</td>
<td>hk</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>A4</td>
<td>465-510</td>
<td>negativ Kontrolle</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>A5</td>
<td>465-510</td>
<td>goi</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>A6</td>
<td>465-510</td>
<td>hk</td>
<td></td>
</tr>
</tbody>
</table>

```r
> Sample Pref color Efficiency Combined sample and target type
<table>
<thead>
<tr>
<th>Sample position</th>
<th>Pref color</th>
<th>Efficiency</th>
<th>Combined sample and target type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$00FF8000</td>
<td>2</td>
<td>Target Negative</td>
</tr>
<tr>
<td>2</td>
<td>clRed</td>
<td>2</td>
<td>Target Unknown</td>
</tr>
<tr>
<td>3</td>
<td>$0030D700</td>
<td>2</td>
<td>Ref Unknown</td>
</tr>
<tr>
<td>4</td>
<td>clFuchsia</td>
<td>2</td>
<td>Target Negative</td>
</tr>
<tr>
<td>5</td>
<td>clGray</td>
<td>2</td>
<td>Target Unknown</td>
</tr>
<tr>
<td>6</td>
<td>$0012D7FA</td>
<td>2</td>
<td>Ref Unknown</td>
</tr>
</tbody>
</table>
```

For adding this additional information to the `CyclesSet` we provide a method for function `merge`.

```r
> cycData1 <- merge(cycData, samInfo)
> ## Extended pheno data
> phenoData(cycData1)
```
An object of class 'AnnotatedDataFrame'
sampleNames: 1 2 ... 96 (96 total)
varLabels: Sample position Sample name ... Combined sample and target type (10 total)
varMetadata: labelDescription

```r
> head(pData(cycData1))

<table>
<thead>
<tr>
<th>Sample position</th>
<th>Sample name</th>
<th>Program number</th>
<th>Segment number</th>
<th>Replicate of</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Sample_1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>Sample_2</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td>Sample_2</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td>Sample_3</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td>Sample_4</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>A6</td>
<td>Sample_4</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
```

Filter combination Target name Sample Pref color Efficiency
1 465-510 negativ Kontrolle $00FF8000 2
2 465-510 goi clRed 2
3 465-510 hk $0030D700 2
4 465-510 negativ Kontrolle clFuchsia 2
5 465-510 goi clGray 2
6 465-510 hk $0012D7FA 2

Combined sample and target type
1 Target Negative
2 Target Unknown
3 Ref Unknown
4 Target Negative
5 Target Unknown
6 Ref Unknown

```r
> varMetadata(phenoData(cycData1))

<table>
<thead>
<tr>
<th>labelDescription</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample position</td>
</tr>
<tr>
<td>Sample name</td>
</tr>
<tr>
<td>Program number</td>
</tr>
<tr>
<td>Segment number</td>
</tr>
<tr>
<td>Replicate of</td>
</tr>
<tr>
<td>Filter combination</td>
</tr>
<tr>
<td>Target name</td>
</tr>
<tr>
<td>Sample Pref color</td>
</tr>
<tr>
<td>Efficiency</td>
</tr>
<tr>
<td>Combined sample and target type</td>
</tr>
</tbody>
</table>
```
3 read.qPCR

read.qPCR allows the user to read in qPCR data and populate a qPCRBatch R object (see section qPCRBatch) using their own data matrix. The format of the data file should be tab delimited and have the following columns, the first two of which are optional (although they should either be provided together, or not at all):

Well Optional, this represents the position of the detector on a plate. This information, if given, will be used to check the plates are of the same size and will also be used in order to plot a representation of the card to look for spatial effects and other potential problems. Both Well number and Plate ID must be present to enable a plate to be plotted.

Plate Optional, this is an identifier for the plate on which an experiment was performed. It is not possible to have duplicate plate IDs with the same Well number. Neither is it possible to have Plate IDs without Well numbers. Both Well number and Plate ID must be present to enable a plate to be plotted.

Sample The sample being analysed. Each sample must contain the same detectors in order to combine and compare samples effectively and to form a valid expression set matrix.

Detector This is the identifier for the gene being investigated. The Detectors must be identical for each sample.

Cq This is the quantification cycle value for a given detector in the corresponding sample.

The generic function read.qPCR is called to read in the qPCR file. It is similar to the read.affybatch function of the "affy" package, in that it reads a file and automatically populates an R object, qPCRBatch described below. However it is different in that the file is user formatted. In addition, unlike read.affybatch, and also unlike the read.taqman function detailed below, only one file may be read in at a time.

If Well and Plate ID information are given, then these are used to populate the exprs.well.order, a new assayData slot introduced in the qPCRBatch object, as detailed below in section qPCRBatch.

So for the qPCR.example.txt file, in directory exData of this library, which contains Well and Plate ID information, as well as the mandatory Sample, Detector and Cq information, we can read in the data as follows.

> path <- system.file("exData", package = "ReadqPCR")
> qPCR.example <- file.path(path, "qPCR.example.txt")
> qPCRBatch.qPCR <- read.qPCR(qPCR.example)
qPCRBatch.qPCR will be a qPCRBatch object with an exprs and exprs.well.order, as well as a phenoData slot which gets automatically populated in the same way as when using read.affybatch. More detail is given in the qPCRBatch section below.

read.qPCR can deal with technical replicates. If the same detector and sample identifier occurs more than once, the suffix _TechReps.n is concatenated to the detector name, where n in \{1,2,...N \} is the number of the replication in the total number of replicates, N, based on order of appearance in the qPCR data file. So for a qPCR file with 2 technical replicates and 8 detectors per replicate, with one replicate per plate, the detector names would be amended as follows:

```r
> qPCR.example.techReps <- file.path(path, "qPCR.techReps.txt")
> qPCRBatch.qPCR.techReps <- read.qPCR(qPCR.example.techReps)
> rownames(exprs(qPCRBatch.qPCR.techReps))[,1:8]
[1] "gene_aj_TechReps.1" "gene_aj_TechReps.2" "gene_al_TechReps.1"
[4] "gene_al_TechReps.2" "gene_ax_TechReps.1" "gene_ax_TechReps.2"
[7] "gene_bo_TechReps.1" "gene_bo_TechReps.2"
```

The reason for appending the suffix when technical replicates are encountered is in order to populate the exprs and exprs.well.order slots correctly and keep them to the assayData format. It also allows the decisions on how to deal with the analysis and combination of technical replicates to be controlled by the user, either using the "NormqPCR" package, or potentially some other function that takes assayData format R objects as input.

### 4 read.taqman

read.taqman allows the user to read in the data output by the Sequence Detection Systems (SDS) software which is the software used to analyse the Taqman Low Density Arrays. This data consists of the header section, which gives some general information about the experiment, run date etc., followed by the raw Cq values detected by the software, followed by summary data about the experiment. read.taqman is a generic function, and is called in a way similar to the read.affybatch function of the "affy" package.

```r
> taqman.example <- file.path(path, "example.txt")
> qPCRBatch.taq <- read.taqman(taqman.example)
```

Currently the SDS software only allows up to 10 plates to be output onto one file. read.taqman allows any number of SDS output files to be combined to make a single qPCRBatch, as long as they have matching detector identifiers.

```r
> path <- system.file("exData", package = "ReadqPCR")
> taqman.example <- file.path(path, "example.txt")
```
SDS output will not necessarily contain plate identifiers, in which case a numeric identifier will be generated, which will increment for each plate, depending on the order of the plates within the SDS files. This is important for filling the `exprs.well.order` slot of the `qPCRBatch`.

`read.taqman` can also deal with technical replicates. If the same detector and sample identifier occurs more than once, the suffix `_TechRep.n` will be concatenated to the detector name, where `n` in `{1, 2,...N}` is the number of the replication in the total number of replicates `N`, based on the order of occurrence in the `taqman` data file. So for a `taqman` file with 4 technical replicates of 96 detectors per sample, with one sample per plate, the detector names would be amended as follows:

```r
> taqman.example.tech.reps <- file.path(path, "exampleTechReps.txt")
> qPCRBatch.taq.tech.reps <- read.taqman(taqman.example.tech.reps)
> rownames(exprs(qPCRBatch.taq.tech.reps))[1:8]

[1] "ACE.Hs00174179_m1_TechReps.1"
[2] "ACE.Hs00174179_m1_TechReps.2"
[3] "ACE.Hs00174179_m1_TechReps.3"
[4] "ACE.Hs00174179_m1_TechReps.4"
[5] "AT1R.AGTR1..Hs00241341_m1_TechReps.1"
[6] "AT1R.AGTR1..Hs00241341_m1_TechReps.2"
[7] "AT1R.AGTR1..Hs00241341_m1_TechReps.3"
[8] "AT1R.AGTR1..Hs00241341_m1_TechReps.4"
```

As with `read.qPCR`, the motivation for appending the suffix when technical replicates are encountered is in order to populate the `exprs` and `exprs.well.order` slots correctly and keep them to the `assayData` format. Again it allows the decisions on how to deal with the analysis of technical replicates to be controlled by the user, either using the "NormqPCR" package, or otherwise.

## 5 qPCRBatch

`qPCRBatch` is an S4 class, designed to store information on the raw Cq values which represents the relative gene expression for a given sample, phenotypic information on the different samples which enable the user to compare expression across different conditions or cell lines, and information on the spatial location of the different detectors used to measure Cq. This is achieved by making `qPCRBatch` an an extension of eSet, which means we
can recycle slots such as exprs and pData, and by introducing a new assyData slot. Here is an example of what a qPCRBatch looks like. note the similarity to eSet:

```r
> qPCRBatch.taq

qPCRBatch (storageMode: lockedEnvironment)
assayData: 96 features, 8 samples
  element names: exprs, exprs.well.order
protocolData: none
phenoData
  sampleNames: fp1.day3.v fp2.day3.v ... fp.8.day.3.mia (8 total)
  varLabels: sample
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation:

pData will be filled automatically if no data is given, in a way analagous to read.affybatch:

```r
> pData(qPCRBatch.taq)

```

In addition there is a new slot, exprs.well.order which extends the assayData slot used for exprs(). It has the same dimensions as exprs (as every instance of assayData must). The cells contain further details on the position on the arrays where the different measurements were taken.

The data provided by this slot can be used in order to identify certain problems with arrays, perhaps due to spatial effects and other problems with the microfluidics technology that is used by many of these systems.

This is conceptually similar to the cdf file information being stored in the AffyBatch class, which contains information on the spatial layout of features on an affy chip. However it differs since it allows for different arrays within the same affyBatch object to have different layouts to each other. This information can be viewed using the exprs.well.order() function.
When using `read.taqman`, if the input file includes identifiers for the different arrays in the experiment, the identifiers will be of the format `<plate.id>-<plate.position>`. However, if no names are given for the different plates, "ReadqPCR" will assign them a numeric identifier, which increments depending on the order of plates in the original file. When several input files are given, as in the case of SDS files, the order in which they are supplied as arguments to the `read.taqman` function will be mirrored in the order of the numeric identifiers for the different plates. However, to minimise confusion, we recommend the user giving the plates their own unique identifiers where possible.

Without plate names:

```r
> head(exprs.well.order(qPCRBatch.taq))

   fp1.day3.v fp2.day3.v fp5.day3.mia fp6.day3.mia
Actb.Rn00667869_m1   "1-38"   "1-134"   "1-230"   "1-326"
Adipoq.Rn00595250_m1  "1-61"   "1-157"   "1-253"   "1-349"
Adrbk1.Rn00562822_m1  "1-66"   "1-162"   "1-258"   "1-354"
Agtr1l.Rn00580252_m1  "1-91"   "1-187"   "1-283"   "1-379"
Alpl.Rn00564931_m1    "1-10"   "1-106"   "1-202"   "1-298"
B2m.Rn00560865_m1     "1-37"   "1-133"   "1-229"   "1-325"
```

With plate names:

```r
> taqman.example.plateNames <- file.path(path, "exampleWithPlateNames.txt")
> qPCRBatch.taq.plateNames <- read.taqman(taqman.example.plateNames)
> head(exprs.well.order(qPCRBatch.taq.plateNames))

   gp1.day3.v gp2.day3.v gp5.day3.mia gp6.day3.mia
Actb.Rn00667869_m1  "PlateA-38"  "PlateA-134"  "PlateA-230"  "PlateA-326"
Adipoq.Rn00595250_m1 "PlateA-61"  "PlateA-157"  "PlateA-253"  "PlateA-349"
Adrbk1.Rn00562822_m1 "PlateA-66"  "PlateA-162"  "PlateA-258"  "PlateA-354"
Agtr1l.Rn00580252_m1 "PlateA-91"  "PlateA-187"  "PlateA-283"  "PlateA-379"
Alpl.Rn00564931_m1   "PlateA-10"  "PlateA-106"  "PlateA-202"  "PlateA-298"
B2m.Rn00560865_m1    "PlateA-37"  "PlateA-133"  "PlateA-229"  "PlateA-325"
```

With plate names:

```r
> head(exprs.well.order(qPCRBatch.taq.plateNames))

   gp1.day3.v gp2.day3.v gp5.day3.mia gp6.day3.mia
Actb.Rn00667869_m1  "PlateA-38"  "PlateA-134"  "PlateA-230"  "PlateA-326"
Adipoq.Rn00595250_m1 "PlateA-61"  "PlateA-157"  "PlateA-253"  "PlateA-349"
Adrbk1.Rn00562822_m1 "PlateA-66"  "PlateA-162"  "PlateA-258"  "PlateA-354"
Agtr1l.Rn00580252_m1 "PlateA-91"  "PlateA-187"  "PlateA-283"  "PlateA-379"
Alpl.Rn00564931_m1   "PlateA-10"  "PlateA-106"  "PlateA-202"  "PlateA-298"
B2m.Rn00560865_m1    "PlateA-37"  "PlateA-133"  "PlateA-229"  "PlateA-325"
```

11
In addition, a mixture of files with and without plate identifiers is possible.

```r
> taqman.example <- file.path(path, "example.txt")
> taqman.example.plateNames <- file.path(path, "exampleWithPlateNames.txt")
> qPCRBatch.taq.mixedPlateNames <- read.taqman(taqman.example,
+     taqman.example.plateNames)
> head(exprs.well.order(qPCRBatch.taq.mixedPlateNames))
```

```plaintext
<table>
<thead>
<tr>
<th>fp1.day3.v</th>
<th>fp2.day3.v</th>
<th>fp5.day3.mia</th>
<th>fp6.day3.mia</th>
</tr>
</thead>
</table>
| Actb.Rn00667869_m1 | "1-38" | "1-134" | "1-230" | "1-326"
| Adipoq.Rn00595250_m1 | "1-61" | "1-157" | "1-253" | "1-349"
| Adrbk1.Rn00562822_m1 | "1-66" | "1-162" | "1-258" | "1-354"
| Agtrl1.Rn00580252_s1 | "1-91" | "1-187" | "1-283" | "1-379"
| Alpl.Rn00564931_m1 | "1-10" | "1-106" | "1-202" | "1-298"
| B2m.Rn00560865_m1 | "1-37" | "1-133" | "1-229" | "1-325"

<table>
<thead>
<tr>
<th>gp.3.day.3.v</th>
<th>gp.4.day.3.v</th>
<th>gp.5.day.3.mia</th>
<th>gp.6.day.3.mia</th>
</tr>
</thead>
</table>
| Actb.Rn00667869_m1 | "2-38" | "2-134" | "2-230" | "2-326"
| Adipoq.Rn00595250_m1 | "2-61" | "2-157" | "2-253" | "2-349"
| Adrbk1.Rn00562822_m1 | "2-66" | "2-162" | "2-258" | "2-354"
| Agtrl1.Rn00580252_s1 | "2-91" | "2-187" | "2-283" | "2-379"
| Alpl.Rn00564931_m1 | "2-10" | "2-106" | "2-202" | "2-298"
| B2m.Rn00560865_m1 | "2-37" | "2-133" | "2-229" | "2-325"

<table>
<thead>
<tr>
<th>gp.3.day.3.v</th>
<th>gp.4.day.3.v</th>
<th>gp.5.day.3.mia</th>
<th>gp.6.day.3.mia</th>
</tr>
</thead>
</table>
| Actb.Rn00667869_m1 | "PlateA-38" | "PlateA-134" | "PlateA-230" | "PlateA-326"
| Adipoq.Rn00595250_m1 | "PlateA-61" | "PlateA-157" | "PlateA-253" | "PlateA-349"
| Adrbk1.Rn00562822_m1 | "PlateA-66" | "PlateA-162" | "PlateA-258" | "PlateA-354"
| Agtrl1.Rn00580252_s1 | "PlateA-91" | "PlateA-187" | "PlateA-283" | "PlateA-379"
| Alpl.Rn00564931_m1 | "PlateA-10" | "PlateA-106" | "PlateA-202" | "PlateA-298"
| B2m.Rn00560865_m1 | "PlateA-37" | "PlateA-133" | "PlateA-229" | "PlateA-325"

<table>
<thead>
<tr>
<th>gp.3.day.3.v</th>
<th>gp.4.day.3.v</th>
<th>gp.5.day.3.mia</th>
<th>gp.6.day.3.mia</th>
</tr>
</thead>
</table>
| Actb.Rn00667869_m1 | "PlateB-38" | "PlateB-134" | "PlateB-230" | "PlateB-326"
| Adipoq.Rn00595250_m1 | "PlateB-61" | "PlateB-157" | "PlateB-253" | "PlateB-349"
| Adrbk1.Rn00562822_m1 | "PlateB-66" | "PlateB-162" | "PlateB-258" | "PlateB-354"
| Agtrl1.Rn00580252_s1 | "PlateB-91" | "PlateB-187" | "PlateB-283" | "PlateB-379"
| Alpl.Rn00564931_m1 | "PlateB-10" | "PlateB-106" | "PlateB-202" | "PlateB-298"
| B2m.Rn00560865_m1 | "PlateB-37" | "PlateB-133" | "PlateB-229" | "PlateB-325"
```

12
If the files to be combined do not have matching detector names, or if duplicate sample or plate names are given, read.taqman will stop and give an error message. When reading in qPCR files with read.qPCR, exprs.well.order will be populated as long as Well and Plate ID columns are given in the input file, otherwise the exprs.well.order slot will be NULL.

So when plate ID and Well data are given:

```r
> head(exprs.well.order(qPCRBatch.qPCR))
```

<table>
<thead>
<tr>
<th></th>
<th>caseA</th>
<th>caseB</th>
<th>controlA</th>
<th>controlB</th>
</tr>
</thead>
<tbody>
<tr>
<td>gene_ai_TechReps.1</td>
<td>A-18</td>
<td>A-43</td>
<td>B-18</td>
<td>B-43</td>
</tr>
<tr>
<td>gene_ai_TechReps.2</td>
<td>C-18</td>
<td>C-43</td>
<td>D-18</td>
<td>D-43</td>
</tr>
<tr>
<td>gene_az_TechReps.1</td>
<td>A-23</td>
<td>A-48</td>
<td>B-23</td>
<td>B-48</td>
</tr>
<tr>
<td>gene_az_TechReps.2</td>
<td>C-23</td>
<td>C-48</td>
<td>D-23</td>
<td>D-48</td>
</tr>
<tr>
<td>gene_bc_TechReps.1</td>
<td>A-6</td>
<td>A-31</td>
<td>B-6</td>
<td>B-31</td>
</tr>
<tr>
<td>gene_bc_TechReps.2</td>
<td>C-6</td>
<td>C-31</td>
<td>D-6</td>
<td>D-31</td>
</tr>
</tbody>
</table>

And when they are not:

```r
> qPCR.example.noPlateOrWell <- file.path(path, "qPCR.noPlateOrWell.txt")
> qPCRBatch.qPCR.noPlateOrWell <- read.qPCR(qPCR.example.noPlateOrWell)
> exprs.well.order(qPCRBatch.qPCR.noPlateOrWell)
```

NULL

Once a qPCRBatch has been populated it is theoretically possible to use any tool which takes as it’s input an exprs set matrix. However it is important to bear in mind the values are not raw expression values but Cq values, and a lower Cq will indicate a higher expression level for a given transcript in the sample. Also it is important to note that when normalising, the amount is relative and is intended to be compared to another condition or tissue type in order to look for differential expression between condition; the technology is not designed to give absolute quantification.