R as Platform for the Analysis of dPCR and qPCR Experiments

by Stefan Rödiger, Michał Burdukiewicz, Konstantin Blagodatskikh and Peter Schierack

Abstract There is an ever-increasing number of applications, which use quantitative PCR (qPCR) or digital PCR (dPCR) to elicit fundamentals of biological processes. Novel amplification strategies based on quantitative isothermal amplification (qIA) become more prominent in life sciences and diagnostics. Several software solutions have been developed, which are either distributed as closed source software or as monolithic block with little freedom to perform highly customized analysis procedures. Others and we argue that R is an excellent environment for a reproducible and transparent analysis of data. However, for newcomers it is often very challenging to master R or learn capabilities of less known packages. Here we describe exemplary workflows for the analysis of dPCR, qIA or qPCR experiments including the analysis of melting curve data. Our analysis relies entirely on R packages available from public repositories.

Introduction

The qPCR is the method of choice when a precise quantification of minute DNA traces of pathogens or the analysis of gene expression is required (Peirson et al., 2003). Numerous commercial and experimental monitoring platforms have been developed in the past years. This includes standard plate cyclers, capillary cyclers, microfluidic platforms and related technologies (Rödiger et al., 2013b; Devonshire et al., 2013; Viturro et al., 2014; Rödiger et al., 2014; Khodakov and Ellis, 2014). Only few bioanalytical applications had such a significant impact on the progress of life sciences and medical sciences as the quantitative Polymerase Chain Reaction (qPCR). The scientific community work hard in the past two decades to uncover pitfalls of qPCR experiments. This lead finally to the development of peer-reviewed analysis algorithms (Ruijter et al., 2013), throughout analysed qPCR chemistries (Ruijter et al., 2014) and guidelines for a proper conduct of qPCR experiments as implemented in the MIQE guidelines (minimum information for publication of quantitative real-time PCR experiments) (Bustin et al.; Huggett et al., 2013). In the past decades emerged several isothermal amplification technologies, such as helicase dependent amplification (HDA), as alternative to PCR. Isothermal amplification was readily combined with real-time monitoring technologies (qIA) and is used nowadays in various fields like diagnostics and point of care testing (Rödiger et al., 2014).

Digital PCR (dpcR) is a novel approach for detection and quantification of nucleic acids. The dPCR Technology breaks fundamentally with the previous concept of nucleic acid quantification and can be seen as a next generation nucleic acid quantification. The key difference between dPCR and traditional PCR lies in the method of measuring (absolute) nucleic acids amounts, which yields discrete information instead of the continuous signal. This is possible after "clonal DNA amplification" in thousands of small separated partitions (e.g., droplets, nano chambers) (Huggett et al., 2013; Milbury et al., 2014; Morley, 2014). Partitions with no nucleic acid remain negative and the others turn positive (e.g., Figure 1). Selected technologies (e.g., OpenArray®Real-Time PCR System) monitor amplification reactions in the chambers in real-time. Cq values are calculated from the amplification curves and converted into discrete events by means of positive and negative partitions and the absolute quantification of nucleic acids is done using Poisson distribution. Recently we published the **dpcR** package at CRAN, which is the first open source software package based on R for the analysis of digital PCR (dPCR) experiments (see **dpcR** for details).

Most of the commercial and experimental hardware platforms provide means to analyse the amplification curve data. Yet, in case of closed source software the analysis happens in most cases in a blackbox fashion tied to a specific platforms. Often such systems have limitation in the data processing and force the used to suboptimal analysis algorithm as discussed by Ruijter et al. (2013). The visualization options are usually limited by the software and not in acceptable publication quality. Aside from closed source software, analysis of PCR reactions is generally performed in spreadsheets. This approach of data processing is not advisable for research purposes. Spreadsheets often lack (or do not use) tools to validate input, debug implemented procedures and automatize workflow. These traits make them prone to errors and not well suited for complicated tasks[1].

The complexity of hardware, wetware and software requires expertise to master a technical workflow comprising standards for experimental design, generation and analysis of data, interpretation of results and reporting (Huggett et al., 2014). We argue that blackboxs are not necessarily a bad thing,

[1]For more elaborated critique see http://www.burns-stat.com/documents/tutorials/spreadsheet-addiction/ and McCullough and Heiser (2008).

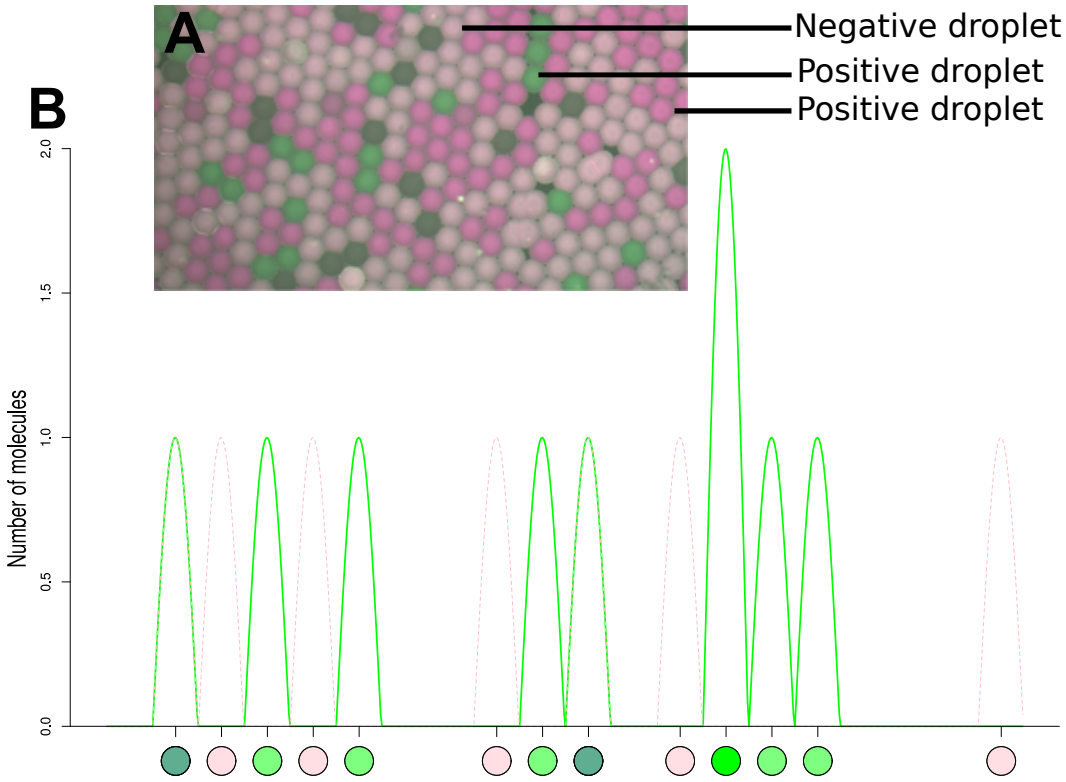

Figure 1: Scheme of droplet digital PCR experiment. **A)** A droplet digital PCR reaction mix was formed in a Bio-Rad QX100 system. The droplets (circa 100 μm in diameter) were subjected to the VideoScan platform for detection and analysis. **(B)** Subsequently the samples can be digitalized by counting number of positive and total number of droplets. The plot was generated with the `sim_ddpcr` function from the **dpcR** package.

but should be avoided wherever possible. Studies by McCullough and Heiser (2008); Almiron et al. (2010); Durán et al. (2014) exemplified this. Scientific misconduct and fraud have shaken the scientific community on several occasions (Fang et al., 2012). In particular qPCR is a sensible topic.

R is one of the most used tools in bioinformatics and is known as an early adopter of emerging technologies (Pabinger et al., 2014) due to several reasons. R provides essential packages to build a highly customized workflows, covering: data read-in, data preprocessing, analysis, post-processing, visualization and storage. As recently briefly reviewed in Pabinger et al. (2014), numerous R packages have been developed for the analysis of qPCR experiments, including: **kulife**, **MCMC.qpcr**, **qPCR.CT**, **DivMelt**, **qpcR**, **dpcR**, **chipPCR**, **MBmca**, **RDML**, **nondetects**, **qpcrNorm**, **HTqPCR**, **SLqPCR**, **ddCt**, **EasyqpcR**, **unifiedWMWqPCR**, **ReadqPCR**, **NormqPCR**. All the packages are either available from CRAN or Bioconductor (Gentleman et al., 2004). The packages can be freely combined in a plugin-like architecture. R is instrument independent, cross-platform and provides a wide spectrum of calculation options. In particular, visualization of experiments is one of R pinnacles. Though the intrinsic properties of R such as the naming convention (Bååth, 2012) and use of R's class systems (e.g., **S3**, **S4**, reference classes and **R6**) vary considerable depending on the package developer preferences there is the common ground to track numerical errors in R due to the open source approach. In addition, offers the R environment several data sets. R offers various methods for a standardized data import/export and exchange. Workflows can be embedded in structures for models (e.g., Predictive Model Markup Language (PMML) as proposed by Zeller et al. (2009), open data exchange formats (e.g., XML-based Real-Time PCR Data Markup Language (RDML) (Lefever et al., 2009), binary formats (Michna and Woods, 2013) or tools provided by the R workspace (R Development Core Team, 2012). Therefore, others and we argue that R is suitable for reproducible research (Gesmann and de Castillo, 2011; Murrell, 2012; Gandrud, 2013; Hofmann et al., 2013; Leeper, 2014; Liu and Pounds, 2014). In addition, several software R packages enable an efficient manipulation, restructuring and reshaping of data to make the readily-available for further processing. This is of particular importance on the human to machine interface (Oh, 2014).

The aim of this paper is to show two simple case studies. In particular, we describe how to:

- read-in data from a standardized file format,
- pre-process the amplification curve data,
- calculate specific parameters from the amplification curve data,
- calculate the melting temperature,
- and report the data.

We share the philosophy of the MIQE guidelines to increase experimental transparency for better experimental practice and reliable interpretation of qPCR results and attempts for open data exchange formats like RDML. We see the application of R in line with this. Our workflow effectively follows the principle illustrated in Figure 2. The intent is to aggregate functionalities dispersed between various packages and offer a fast insight for novices in the analysis of qPCR experiments with R.

Setting-up a working environment

We recommend to perform the scripting in a dedicated integrated development environment (IDE) and graphical user interface (GUI) such as **RKWard** (Rödiger et al., 2012), **Rstudio**[2] or related technologies (Valero-Mora and Ledesma, 2012). Benefits of IDE's with GUI include syntax-highlighting, auto completion and function references for rapid prototyping of workflows.

Typically the qPCR analysis will start with data from a commercial platform. Most platforms have an option to export a CSV file or spreadsheets application file (e.g., *.xls, *.odt). The details for the import data has been described elsewhere (R Development Core Team, 2012; Rödiger et al., 2012). To keep the case study sections compact we have chosen to load datasets from the **qcpR** package (Ritz and Spiess, 2008; Spiess et al., 2008) (v. 1.4.0) and **RDML** package to our workspace. In this study we used the **RDML** package (v. 0.4-2) for data read-in. The data were measured with a CFX96 System (Bio-Rad) and then exported as RDML v1.1 format file. The **chipPCR** package (v. 0.0.8-3) was used for data preprocessing, quality control and the calculation of the quantification cycle (Cq). The Cq is a quantitative measure, which represents the number of cycles needed to reach a user defined threshold fluorescence signal level. Typically Cq are determined identically the exponential phase of a qPCR reaction. Several Cq methods have been described (Ruijter et al., 2013). In this study we have chosen the second derivative maximum method (Cq_{SDM}) and the "Cycle threshold" (Cq_{Ct}) method.

In a perfect qPCR reaction, the amount of amplicon doubles (2^n; n = cycle number) at each cycle. Here the amplification efficiency (AE) is 100 %. However, in reality, numerous factors cause an

[2] http://www.rstudio.com/

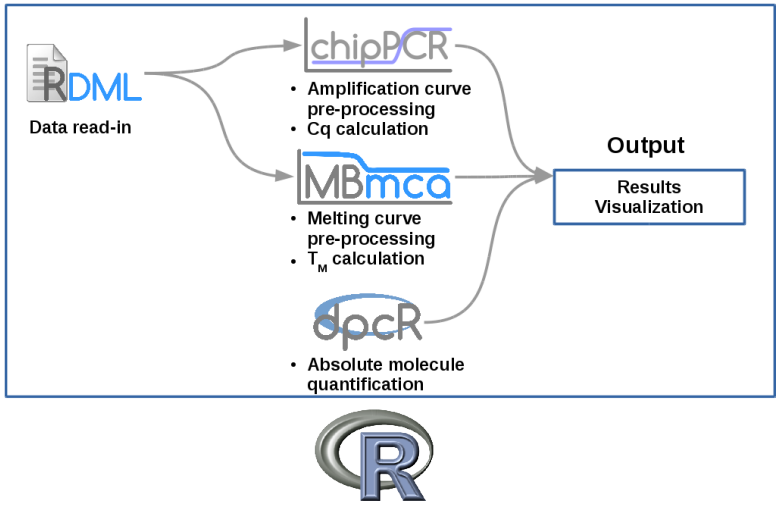

Figure 2: Exemplary workflow for qPCR experiments in R. Core functionality is provided by the R software environment for statistical computing and graphics. In our scenario we used the **RDML** package to read-in data in standardized format. However, any format supported by R could be used. Further processing of amplification curve data was performed with the **chipPCR** package and amplification curve data were analysed with the **MBmca** package. Cq, quantification cycle; T_M, melting temperature. The **dpcR** package can be embedded in the analysis of digital PCR experiments.

inhibition of the amplification ($AE < 100$ %). The AE can determined be the relation of the Cq value depending on the sample input quantity as detailed described in (Rödiger and Burdukiewicz, 2014).

In Rödiger et al. (2013a) we described the application of R for the analysis of melting curve experiments on the surface of microbeads. Since the mathematical foundation for melting curve analysis (MCA) is identical between all platforms we applied the functions from the **MBmca** package (Rödiger et al., 2013a) for an analysis of the target specific melting temperature (T_M) in our qPCR experiment. We used the **MBmca** package (0.0.3-4) for analysis of melting curve data.

We complete our study with a simple case study for the analysis of dPCR experiment. In particular, we used the **dpcR** (0.1.3.1) to estimate the number of molecules in a sample.

Results

In this section we will try to show that R is a unified open software which fits the needs for (I) data analysis and presentation in research, (II) as software frame-work for novel technical developments, (III) as platform for teaching this new technology and (IV) serves as reference for statistical methods.

Case study one – qPCR and Amplification Efficiency Calculation

The goal of our fist case study was to calculate the Cq values and the AE from a qPCR experiment. We used the "guescini1" dataset from the **qcpR** package. Details of the experiment are described in Guescini et al. (2008). First we start with loading the required packages and datasets. A good practice for reproducible research is to track the package versions and environment used during the analysis. The function $sessionInfo()$ from the **utils** package provides this information. Assuming that the analysis starts with a clean R session it is possible to assign the required packages to an object only, as shown in our case study below[3].

```
# Load the required packages for the data import and analysis.
# Load the chipPCR package for the pre-processing and curve data quality
# analysis and load the qpcR package as data resource.
require(chipPCR)
require(qpcR)

# Collect information about the R session used for the analysis of the
```

[3]The reproducibility of research can be further improved by using dedicated tools. For example, **archivist** package allows not only stores and recovers crucial data, but preserves metadata of saved objects. All settings of R session can be easily saved and/or restored using **settings** package.

```
# experiment.
current.session <- sessionInfo()

# Next we load the 'guescini1' dataset from the qpcR package the to
# workspace and assign it to the object gue.
gue <- guescini1

# Define the dilution of the sample DNA quantity for the calibration curve.
dil <- 10^(2:-4)
```

Next, we previewed the amplification curve raw data using the `matplot` function (see code below). The amplification curve data showed a strong signal level variation in the plateau region (Figure 3A). Therefore all data were subjected to a minimum-maximum normalization (compare Rödiger et al. (2013a)) using the `CPP` function from the **chipPCR** package. In addition, all data were baselined and smoothed (Figure 3B). The Cq values were calculated by the Cq_{SDM} and Cq_{Ct} methods.

```
# Pre-process the amplification curve data with the CPP function from the
# chipPCR package. The trans parameter was set TRUE to perform a baselining and
# the method.norm parameter was set to minm for a min-maximum normalization.

res.CPP <- cbind(gue[, 1], apply(gue[, -1], 2, function(x) {
  CPP(gue[, 1], x, trans = TRUE, method.norm = "minm",
      bg.range = c(1,7))[["y.norm"]]
}))

# Use the th.cyc function from the chipPCR package to calculate the Cq values
# by the cycle threshold method at a threshold level "r" of 0.05.
Cq.Ct <- apply(gue[, -1], 2, function(x)
  th.cyc(res.CPP[, 1], x, r = 0.05)[1])

Cq.SDM <- apply(gue[, -1], 2, function(x)
  summary(inder(res.CPP[, 1], x))[2])

# Fit a linear model to carry out a regression analysis.
res.Cq <- lm(Cq.Ct ~ Cq.SDM)

summary(res.Cq)
```

The dilution "dil" and the "Cq.Ct" values served as input for the calculations. In our case study we needed to rearrange to "Cq.Ct" values in a matrix. For visualization of the confidence intervals of the regression analysis we set the parameter $CI = TRUE$.

```
> summary(res.Cq)

Call:
lm(formula = Cq.Ct ~ Cq.SDM)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4904 -0.2730  0.0601  0.3540  1.1871

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -8.125534   0.207419  -39.17   <2e-16 ***
Cq.SDM       0.988504   0.008097  122.08   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5281 on 82 degrees of freedom
Multiple R-squared:  0.9945,        Adjusted R-squared:  0.9945
F-statistic: 1.49e+04 on 1 and 82 DF,  p-value: < 2.2e-16

layout(matrix(c(1,2,3,3,4,5), 3, 2, byrow = TRUE))

matplot(gue[, -1], type = "l", lty = 1, col = 1, xlab = "Cycle",
```

```
            ylab = "RFU", main = "Raw data")
legend("topleft", "A", cex = 3, bty = "n")

matplot(res.CPP[, -1], type = "l", lty = 1, col = 1, xlab = "Cycle",
        ylab = "RFU", main = "Pre-processed data")
legend("topleft", "B", cex = 3, bty = "n")
abline(h = 0.05, col = "red", lwd = 2)

plot(Cq.SDM, Cq.Ct, xlab = "Ct method", ylab = "SDM method",
     main = "Comparison of Cq methods")
abline(res.Cq)
legend("topleft", "C", cex = 3, bty = "n")

plot(effcalc(dil, t(matrix(Cq.Ct, nrow = 12, ncol = 7))), CI = TRUE)
legend("topright", "D", cex = 3, bty = "n")

plot(effcalc(dil, t(matrix(Cq.SDM, nrow = 12, ncol = 7))), CI = TRUE)
legend("topright", "E", cex = 3, bty = "n")
```

Case study two – qPCR and Melting Curve Analysis

A common task during the analysis of qPCR experiments is to distinguish between positive and negative samples (compare Figure 4). Provided that the melting temperature of a sample is known it is possible to automatize the melting curve analysis (MCA). As shown in Rödiger et al. (2013a) this can be done by interrogating the T_M. Therefore, we used a simple logic statement, which checks if T_M is within a tight temperature range and signal height. In line with "Case study one" we used function $sessionInfo()$ to track the packages used for the analysis. Reproducible research is greatly enhanced of open data exchange formats are used. In this study we used the **RDML** package to read the qPCR experiment. The example file 'BioRad_qPCR_melt.rdml' for analysis was taken from the **RDML** package. Within this qPCR experiment we amplified the *Mycobacterium tuberculosis katG*[4] gene to detect mutation at codon 315. The experiment was separated for two parts: (I) detection of overall *M.tuberculosis* DNA (wild-type and mutant) by intercalating dye EvaGreen® within qPCR and (II) specific detection of wild-type *M.tuberculosis* by melting of TaqMan probe (quencher – BHQ2, fluorescent reporter – Cy5) with amplified DNA (similar assay is described in Luo et al. (2011)). Real time amplification were conducted using Syntol EvaGreen Master Mix according to the manufacturer's instructions, with 500 nM of primers and probe in a 25 μL final reaction volume. Thermocycling was conducted using a CFX96 (BioRad) initiated by 3 min incubation at 95 °C, followed by 41 cycles (15 s at 95 °C; 40 s at 65 °C) with a single fluorescent reading taken at the end of each cycle. Probe melting was conducted between 35 °C and 95 °C by 1 °Cat 1 s steps.

```
# Load the required packages for the data import and analysis.

# Import the qPCR and melting curve data via the RDML package
require(RDML)

# Load the chipPCR package for the pre-processing and curve data quality
# analysis.
require(chipPCR)

# Load the MBmca package for the melting curve analysis.
require(MBmca)

# Collect information about the R session used for the analysis of the qPCR
# experiment.
current.session <- sessionInfo()


# Load the BioRad_qPCR_melt.rdml file form RDML package and assign the data to
the
# object BioRad.

filename <- paste(path.package("RDML"), "/extdata/", "BioRad_qPCR_melt.rdml",
```

[4] https://www.wikigenes.org/e/gene/e/923602.html

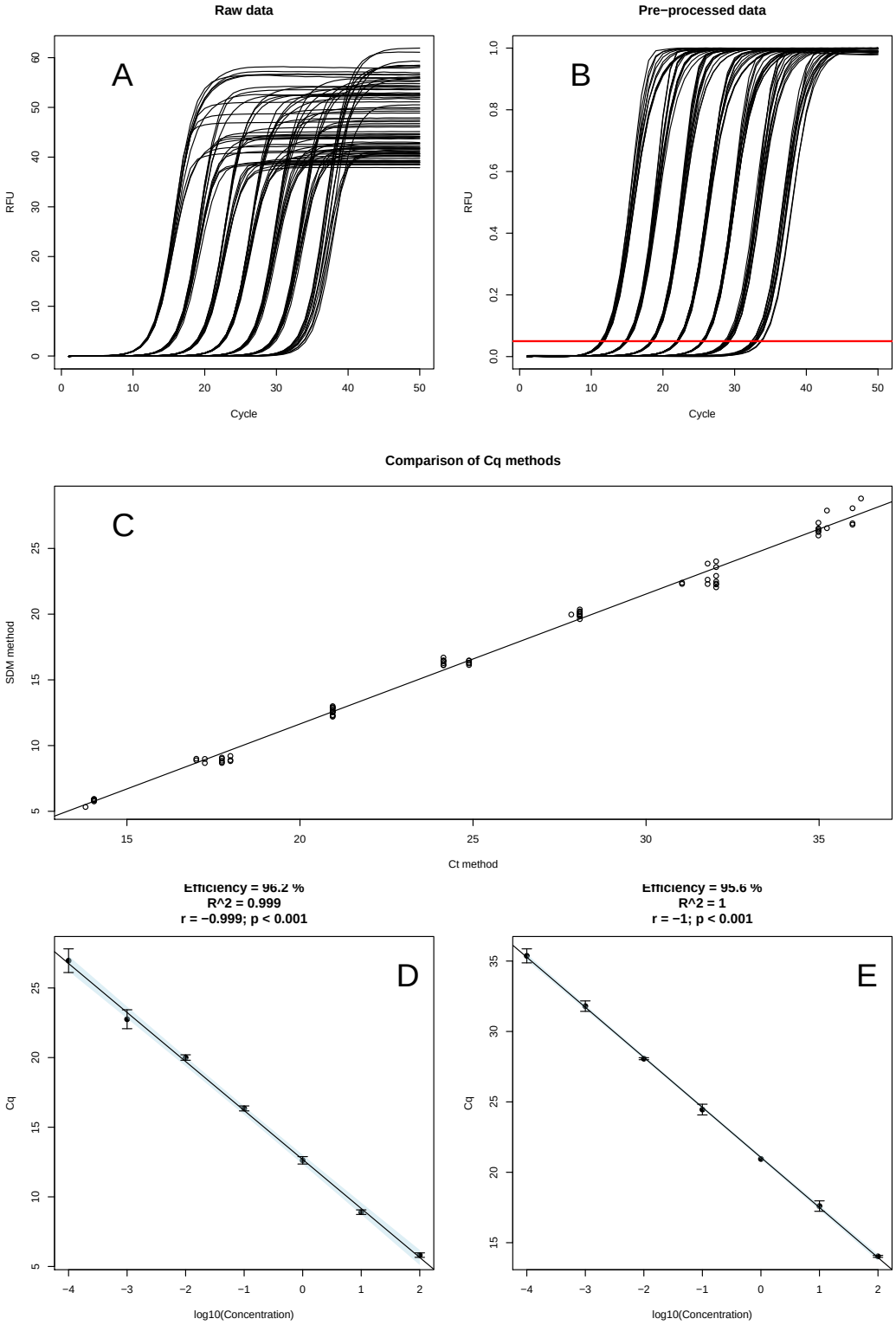

Figure 3: Analysis of the amplification curve data of the calibration curve samples. **(A)** Visual inspection of the raw data from the `guescini1` dataset. The qPCR curves display a broad variation in plateau fluorescence (38 – 62 RFU). The red horizontal line indicates the fluorescence level (0.05) used for the calculation of the Cq by the "cycle threshold" method. **(B)** the `CPP` function from the **chipPCR** was sued to baseline the data, to smooth the data with Savitzky-Golay smoothing filter and to normalize the data between 0 and 1. **(C)** The Cq values were calculated for the second derivative maximum (Cq_{SDM}) method (`inder`, **chipPCR**) and the cycle threshold method (Cq_{Ct}) (`th.cyc`, **chipPCR**). The threshold value was set to $r = 0.05$. The Cq_{SDM} and Cq_{Ct} values were plotted and analysed by a linear regression ($R^2 = 0.9945$; $P < 2.2^{-16}$) and Pearson's ($r = 0.9972605$; $P < 2.2^{-16}$). The AE based on **(D)** Cq_{Ct} values and **(E)** Cq_{SDM} values were automatically analysed with the `effcalc` (**chipPCR**) function.

```
sep = "")
BioRad <- RDML(filename, name.pattern = "%TUBE%_%NAME%_%TYPE%_%TARGET%")

# Fetch cycle dependent fluorescence for the EvaGreen channel of the
# Mycobacterium tuberculosis katG gene and aggregate the data in the
# object qPCR.
qPCR <- cbind(BioRad[["qPCR"]][["EvaGreen"]][["pos"]],
              BioRad[["qPCR"]][["EvaGreen"]][["unkn"]][, -1],
              BioRad[["qPCR"]][["EvaGreen"]][["ntc"]][, -1])
# Leave data only from row 'D' that contains target 'Cy5-2' at channel 'Cy5'
qPCR <- cbind(qPCR[,1], qPCR[, grep("^D", names(qPCR))])
```

We inspected and pre-processed a subset of the amplification curve data solely using functionalities provided by the **chipPCR** package. The `plotCurves` function was used to get an overview of the curvatures. The data indicated a baseline shift in all curves with a slight negative trend (Figure 4). This suggested to baseline the raw data by using a linear regression model (cycles x - y; ($bg.range = c(x, y)$) in the `CPP` function.).

```
# Use plotCurves function from the chipPCR package to get an overview of the
# amplification curve samples.

plotCurves(qPCR[, 1], qPCR[, -1], type = "l")

# Detect positive samples - calculate Cq values
# by the cycle threshold method. The threshold level r was set to 50.
Cq.Positive <- t(apply(qPCR[, -1], 2, function(x)
{
  res <- CPP(qPCR[, 1], x, trans = TRUE, bg.range = c(2, 8))[["y.norm"]]
  th.cyc <- th.cyc(qPCR[, 1], res, r = 50)[1]
  cq <- as.numeric(th.cyc)
  pos <- !is.na(cq)
  c(Cq=cq, M.Tub_positive = pos)
}
))
Cq.Positive
```

Since the qPCR indicated that selected samples (except no template control) are positive, we distinguished between true positive and true negative samples by MCA.

```
# Fetch temperature dependent fluorescence for the Cy5 channel of the
# probe that can hybridize with Mycobacterium tuberculosis katG gene (codon 315)
# and aggregate the data in the object melt.
melt <- cbind(BioRad[["Melt"]][["Cy5-2"]][["pos"]],
              BioRad[["Melt"]][["Cy5-2"]][["unkn"]][, -1],
              BioRad[["Melt"]][["Cy5-2"]][["ntc"]][, -1])

# Calculate the melting temperature with the diffQ function
# from the MBmca package. Use simple logical conditions to find out
# if a positive sample with the expected Tm of circa 54.5 degree
# Celsius is found.
Tm.Positive <- matrix(nrow = length(melt[, -1]),
                      byrow = TRUE,
                      dimnames = list(names(melt)[-1]),
                      unlist(apply(melt[, -1], 2, function(x) {
  res.Tm <- diffQ(cbind(melt[, 1], x), fct = max, inder = TRUE)
  positive <- ifelse(res.Tm[1] > 54 & res.Tm[1] < 55 & res.Tm[2] > 80, 1, 0)
  c(res.Tm[1], res.Tm[2], positive)
})))

# Present the results in a tabular output as data.frame "results.tab".
# Result of analysis logic is:
# Cq.Positive && Tm.Positive = Wild-type
# Cq.Positive && !Tm.Positive = Mutant
# !Cq.Positive && !Tm.Positive = NTC
# !Cq.Positive && Tm.Positive = Error
results <- sapply(1:length(Cq.Positive[,1]), function(i) {
```

```
  if(Cq.Positive[i, 2] == 1 && Tm.Positive[i, 3] == 1)
    return("Wild-type")
  if(Cq.Positive[i, 2] == 1 && Tm.Positive[i, 3] == 0)
    return("Mutant")
  if(Cq.Positive[i, 2] == 0 && Tm.Positive[i, 3] == 0)
    return("NTC")
  if(Cq.Positive[i, 2] == 0 && Tm.Positive[i, 3] == 1)
    return("Error")
})
results.tab <- data.frame(cbind(Cq.Positive, Tm.Positive, results))
names(results.tab) <- c("Cq", "M.Tub DNA", "Tm", "Height", "Tm positive",
"Result")
results.tab[["M.Tub DNA"]] <- factor(results.tab[["M.Tub DNA"]], labels=c("Not
Detected",
                                                                    "Detected"))
results.tab[["Tm positive"]] <- factor(results.tab[["Tm positive"]],
labels=c(TRUE,
                                                                    FALSE))
results.tab
```

Finally, we plotted and printed the output of our analysis.

```
# Convert the decision from the "relsults" object in a color code:
# Negative, black; Positive, red.

color <- c(Tm.Positive[, 3] + 1)

# Arrange the results of the calculations in plot.
layout(matrix(c(1,2,1,3), 2, 2, byrow = TRUE))

# Use the CPP function to preporcess the amplification curve data.
plot(NA, NA, xlim = c(1, 41), ylim = c(0,200), xlab = "Cycle", ylab = "RFU")
mtext("A", cex = 2, side = 3, adj = 0, font = 2)
lapply(2L:ncol(qPCR), function(i)
  lines(qPCR[, 1],
        CPP(qPCR[, 1], qPCR[, i], trans = TRUE,
            bg.range = c(1,9))[["y.norm"]],
        col = color[i - 1]))
matplot(melt[, 1], melt[, -1], type = "l", col = color,
        lty = 1, xlab = "Temperature [Â°C]", ylab = "RFU")
mtext("B", cex = 2, side = 3, adj = 0, font = 2)

plot(NA, NA, xlim = c(35, 95), ylim = c(-15, 120), xlab = "Temperature [Â°C]",
     ylab = "-d(RFU)/dT")
mtext("C", cex = 2, side = 3, adj = 0, font = 2)
lapply(2L:ncol(melt), function(i)
  lines(diffQ(cbind(melt[, 1], melt[, i]), verbose = TRUE,
              fct = max, inder = TRUE)[["xy"]], col = color[i - 1]))
```

Case study three – Isothermal Amplification

Isothermal amplification is an alternative to PCR. In contrast to PCR use isothermal amplification methods a constant temperature rather than cycling through denaturation, annealing and extension steps. The corresponding signal is monitored depending on the time instead of cycles (Rödiger et al., 2014). We performed a quantitative isothermal amplification (qIA) for the target *pCNG1*[5] by using a Helicase Dependent Amplification (HDA). The enzyme DNA Helicase unwinds DNA. Therefore, no

[5]The HDA conditions were taken from the "IsoAmp III Universal tHDA Kit", Biohelix Corp, as described by the vendor. In detail, the reaction was composed of "mix A)" 10 μL A. bidest, 1.25 μL 10xb uffer, 0.75 μL primer(150 nM final), 0.5 μL template plasmid. Preincubation: The mixture was incubated for 2 min at 95°Cand immediately placed on ice. Reaction "mix B)" contained 5 μL A. bidest., 1.25 μL 10x buffer, 2 μL NaCl, 1.25 μL $MgSO_4$, 1, 75μL dNTPs, 0.25 μL EvaGreen (Biotium), 1 μL enzyme mix. The mix was covered with 50 μL mineral oil (Roth). The fluorescence measurement in VideoScan HCU started directly after adding "mix B)" at 65°C. A 1x (D1) and a 1 : 10 dilution (D2) were tested.

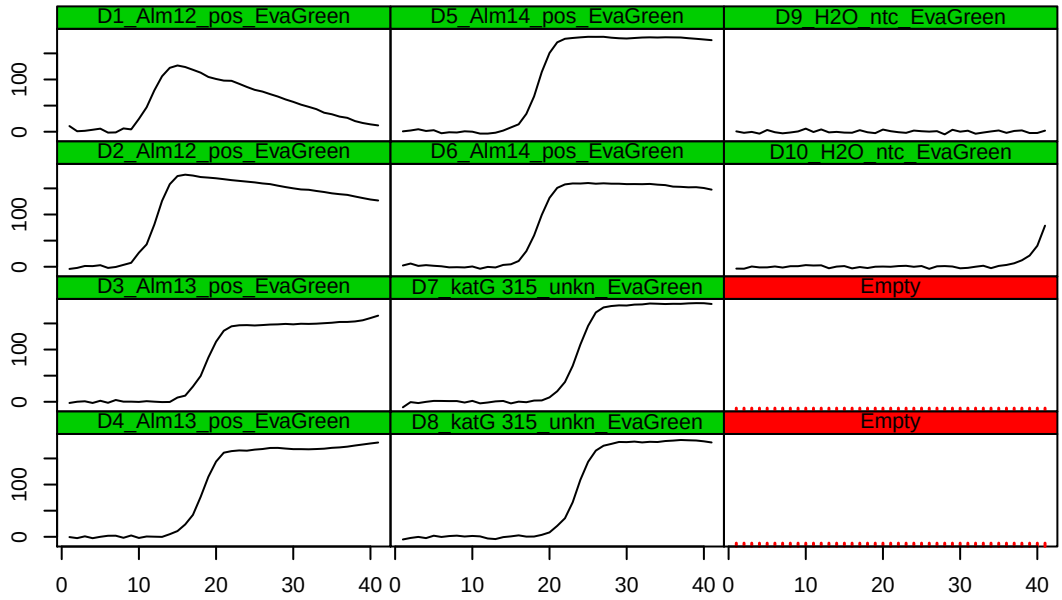

Figure 4: Analysis of the amplification curve data of the calibration curve samples by the `plotCurves` function from the **chipPCR** package. The green color code indicates that the data contains no missing values. However, the visual inspection revealed that the data are noisy. All samples ("D1_Alm12…" - "D8_Alm12…") appeared to be positive. One negative control ("D10_H20_ntc_EvaGreen") appeared to be contaminated.

thermal denaturation is needed. Our previously reported VideoScan platform (Rödiger et al., 2013b) was used to measure the samples. The resulting dataset `C81` is part of the **chipPCR** package. Two concentrations (stock and 1:10 diluted stock) of input DNA were used in the HDA.

```
# Drawn in an 2-by-1 array on the device by two columns and one row.
par(mfrow = c(2, 1))

# Plot the raw data from the C81 dataset to the first array and add
# a legend.
plot(NA, NA, xlim = c(0, 120), ylim = c(0.4, 1.2), xlab = "Time (min)", ylab =
"RFU")
mtext("A", cex = 2, side = 3, adj = 0, font = 2)
lapply(c(2, 4), function(i) {
    lines(C81[, i]/60, C81[, i + 1], type = "b", pch = 20, col = i - 1)
})
legend(10, 0.8, c("D1: 1x", "D2: 1:10 diluted sample"), pch = 19, col = c(1,
3),
    bty = "n")

# Prepare a plot on the second array for the pre-proccessed data.
plot(NA, NA, xlim = c(0, 120), ylim = c(0, 0.8), xlab = "Time (min)", ylab =
"RFU")
mtext("B", cex = 2, side = 3, adj = 0, font = 2)

# Apply the CPP functions to pro-process the raw data.
# 1) Baseline data to zero, 2) Smooth data with spline,
# 3) Remove outliers in background range between
# entry 1 and 190.
res <- lapply(c(2, 4), function(i) {
  y.s <- CPP(C81[, i]/60, C81[, i + 1],
             trans = TRUE,
```

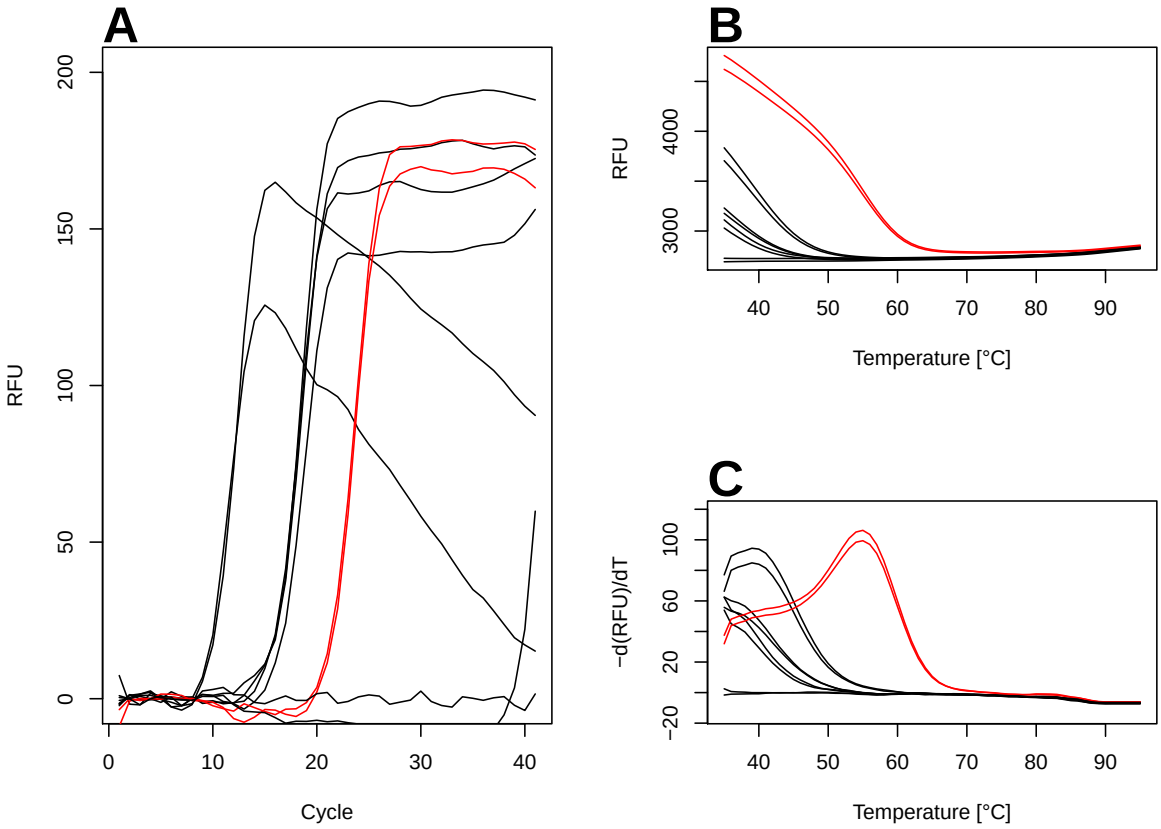

Figure 5: Graphical presentation of the amplification curve data and melting curve data. **(A)** The raw amplification curve data were pre-process with the `CPP` function prior to visualization. To calculate the T_M values from the raw melting curve data **(B)** we used the `diffQ` function from the **MBmca** package. **(C)** We adjusted our algorithm to plot the positive melting peaks in red, while negative melting peaks were labelled in black.

```
            method = "spline",
            bg.outliers = TRUE,
            bg.range = c(1, 190))
  lines(C81[, i]/60, y.s[["y.norm"]], type = "b", pch = 20, col = i - 1)
  # Use the th.cyc function to calculate the cycle threshold time.
  # The threshold level r was set to 0.05.
  paste(round(th.cyc(C81[, i]/60, y.s[["y.norm"]], r = 0.05)[1], 2), "min")
})

# Add the cycle threshold time and the threshold level to plot.

abline(h = 0.05, lty = 2)
text(10, 0.55, "Cq:")
legend(10, 0.5, paste(c("D1: ", "D2: "), res), pch = 19, col = c(1, 3), bty =
"n")
```

Case study four - digital PCR

We have developed the **dpcR** package for analysis and presentation of digital PCR experiments. The **dpcR** package can be used to build custom-made analysers and provides structures to be openly extended by the scientific community. Simulations and predictions of binomial and Poisson distributions, commonly used theoretical models of dPCR, statistical data analysis methods, plotting facilities and report generation tools are part of the package (Pabinger et al., 2014). Here, we show briefly a case study for the **dpcR**.

In detail, we mimicked an in-silico experiment for a droplet digital PCR similar to Figure 1. The aim was to do a density estimation. The number of positive partitions (k), total number of partitions (n) and the size of the partition are only data required for analysis. The estimate of the mean number of template molecules per partition ($\hat{\lambda}$) is calculated using following equation (Huggett et al., 2013):

$$\hat{\lambda} = -\ln(1 - \frac{k}{n}). \tag{1}$$

The average droplet value in our experiment was assumed to be 5 nL. We counted $n = 16800$ droplets in total from which $k = 4601$ droplets were positive. Since the binomial distribution of positive and negative partitions is used to define $\hat{\lambda}$, we use Wilson method for calculation of confidence intervals (Brown et al., 2001). The obtained mean number of template molecules per partition multiplied by the volume of the partitions constitutes the sample concentration.

```
# Load the dpcR package for the analysis of the digital PCR experiment.
require(dpcR)

# In our in-silico experiment we counted in total 16800 droplets (n).
# Thereof, 4601 were positive (k).

(dens <- dpcr_density(k = 4601, n = 16800, average = TRUE, methods = "wilson"))

# Let us assume, that every droplet has roughly 5 nL
# total concentration (and its confidence intervals) in molecules/mL
dens[4:6] / 5 * 1e-6
#results:
#concentration        lower       upper
# 6.400498e-08  6.217049e-08 6.58852e-08
```

Selected functionality was implemented as interactive **shiny** GUI application to make the software accessible for users who are not fluent in R and for experts who wish to automatize routine tasks. Details and examples of the **shiny** web application framework for R can be found at http://shiny.rstudio.com/. We implemented flexible yet simple user interfaces, which run the analyses and graphical representation into interactive web applications either as service on a web severer or on a local machine without knowledge of HTML or ECMAScript (see **dpcR** manual). The interface is designed in a cascade workflow approach (Data import → Analysis → Output → Export) with interactive users choice on input data, methods and parameters using typical GUI elements such as sliders, drop-downs and text fields. An example can be found at https://michbur.shinyapps.io/dpcr_density/. This approach enables the automatized outputs of R objects in combined plots, tables and summaries.

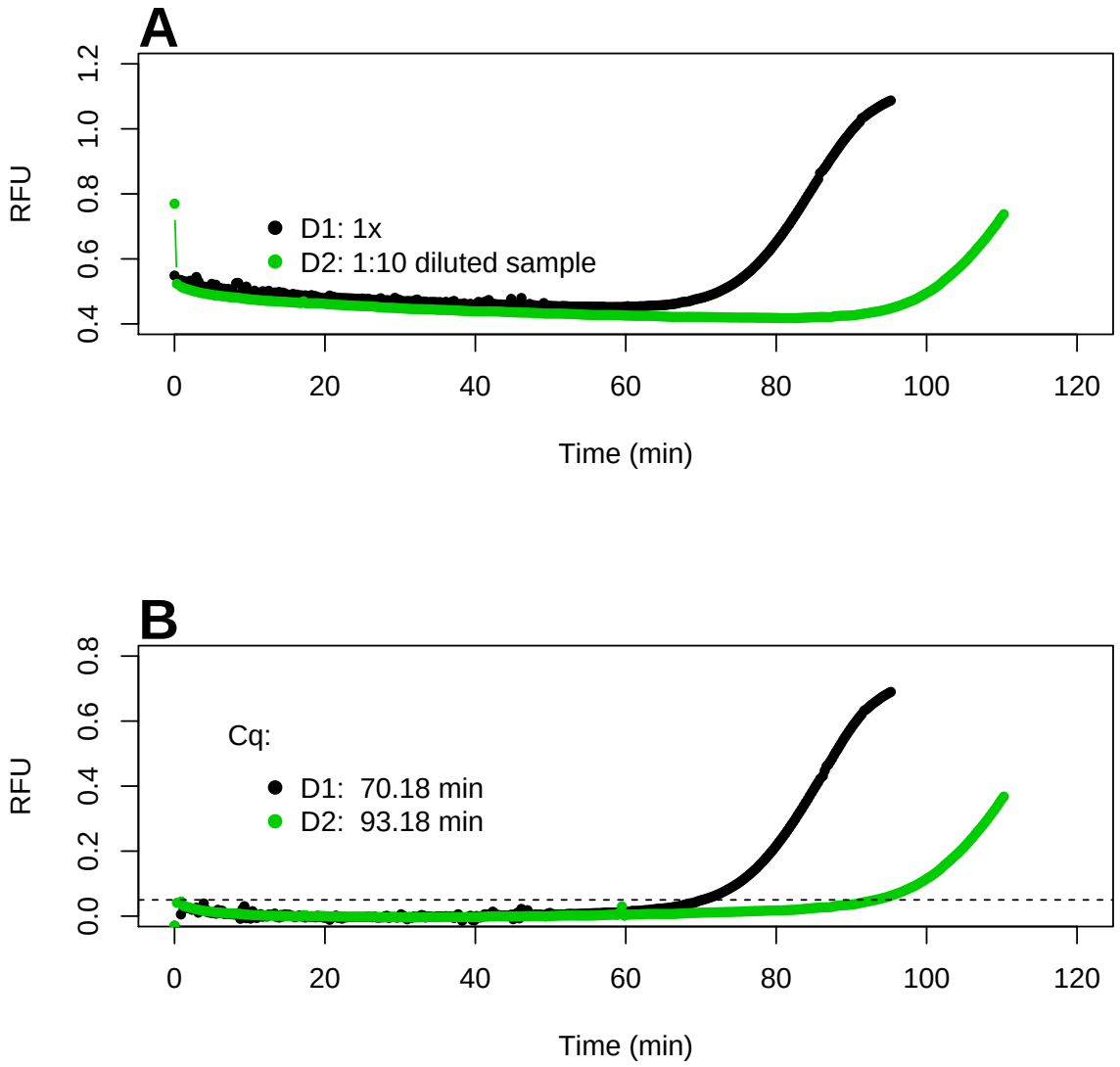

Figure 6: Quantitative isothermal amplification by Helicase Dependent Amplification (HDA). **(A)** The raw data of the HDA (D1, undiluted, D2 1 : 10 diluted) exhibit some outliers (detector artifacts), an off-set of circa 0.5 relative fluorescence units (RFU) and a slight negative trend in the baseline region (0 to 52 minutes). **(B)** First we used the `CPP` function from the **chipPCR** package to smooth the data with spline function. Baselining was done with a robust MM-estimator (range 0 to 52 min). Finally, we used the `th.cyc` function from the **chipPCR** package to calculate the cycle threshold time for samples D1 and D2. The threshold value was set to $r = 0.05$ (−−).

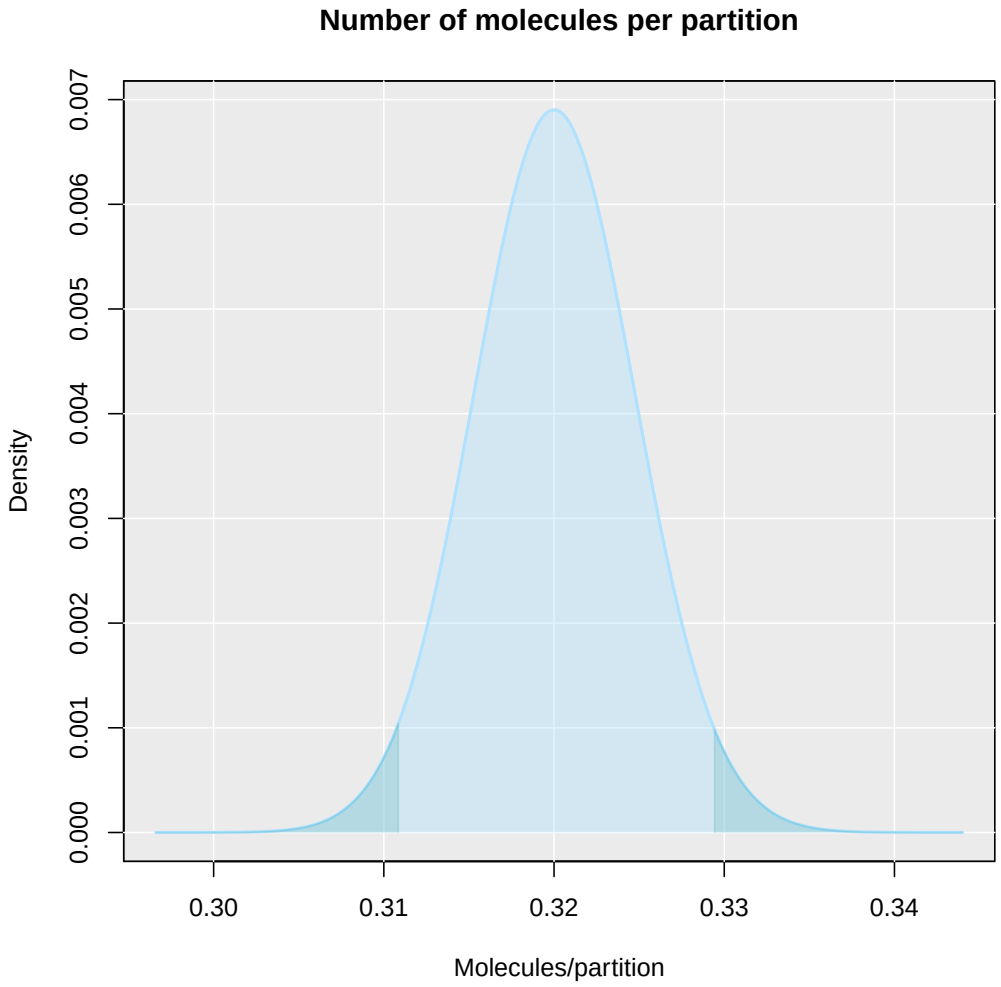

Figure 7: `dpcr_density` function from the **dpcR** package used for analysis of droplet digital PCR experiment. From 16800 counted droplets (n) 4601 were positive (k). The chart presents the distribution of mean number of template molecules per partition ($\hat{\lambda}$).

Discussion and Conclusion

This study gave a brief introduction how to perform a qPCR, qIA or dPCR analysis with R based on packages available from CRAN. In addition, we briefly referenced to a vast collection of additional packages available from CRAN and Bioconductor. The packages may be considered building blocks (libraries) to create what users want and need. We showed that automatized research with R offers powerful means for statistical analysis and visualization. The software is not tied to a vendor or specific application (e.g., chamber or droplet based digital PCR, capillary or plate qPCR). It should be quite easy even for an inexperienced user to define a workflow and to set up environment for specific needs in a broad range of technical settings (Figure 8). R enforces no monolithic integration. We claim that the modular structure of R packages allows user to perform flexible data analysis adjusted to their needs and to design frameworks for high-throughput analysis. R allows to access and reuse code for the creation of reports in various formats (e.g., HTML, PDF). Most of the software is cross-platform open source software and is freely available from CRAN or Bioconductor. Despite the fact that R is free of charge it is quite possible to build commercial applications. The packages cover implementation of novel approaches and peer-reviewed analysis methods. R packages are an open environment to adopt to the growing knowledge in dPCR and qPCR. Therefore, we argue that R may provide a structure for standardized nomenclature and serve as reference in qPCR and dPCR analysis. Speaking about openness, it is important to emphasize that main advantage is the software is transparent at any time for anybody. Thus, it is possible to track numerical errors. A serve disadvantage of R is the lack of comprehensive GUIs for qPCR analysis. Other and we believe that a graphical user interface (GUI) is a key technology to spread the use of R in bioanalytical sciences. Currently, we are establishing the "pcRuniveRsum" (http://michbur.github.io/pcRuniveRsum/) as an on-line resource for interested users. The command-line structure makes R "inaccessible" for many novices. This we support the attempt that automatic routines are made accessible via GUIs (Rödiger et al., 2012). However, work in this has recently started and is still under development.

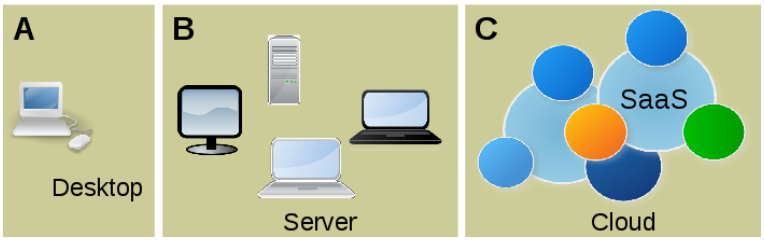

Figure 8: Deployment of R applications for the qPCR and dPCR experiments. **(A)** R is typically run from a desktop computer an operated by an GUI/IDE application such as **Rstudio** or **RKWard**. This approach is provides a flexible workflow for individuals. **(B)** Another approach is to run R with specific applications on a local server. Such scenarios are useful for the deployment within research departments or cooperate units. **(C)** Cloud computing (CC) provides shared and scalable computing capacity (e.g., computing capacity, application software) and storage capacity (e.g., databases) as a service to an individual user or a community Service categories include: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) over a network. Providers of CC manage the infrastructure and resources to achieve coherence and economies of scale similar to a utility over a network (typically the Internet).

Acknowledgment

Part of this work was funded by the BMBF InnoProfile-Transfer-Projekt 03 IPT 611X. We would like to thank R community. Part of this work was funded by the Russian Ministry of Education and Science (project No. RFMEFI62114X0003) and with usage of scientific equipment of Center for collective use "Biotechnology" at All-Russia Research Institute of Agricultural Biotechnology. We would like to thank Mario Menschikowski (Technical University Dresden) for the droplet digital PCR samples.

Bibliography

M. G. Almiron, B. Lopes, A. L. C. Oliveira, A. C. Medeiros, and A. C. Frery. On the numerical accuracy of spreadsheets. *Journal of Statistical Software*, 34(4):1–29, 4 2010. ISSN 1548-7660. URL http://www.jstatsoft.org/v34/i04. [p2]

R. Bååth. The State of Naming Conventions in R. *The R Journal*, 4(2):74–75, Dec. 2012. URL http://journal.r-project.org/archive/2012-2/RJournal_2012-2_Baaaath.pdf. [p2]

L. D. Brown, T. T. Cai, and A. Dasgupta. Interval estimation for a binomial proportion. *Statistical Science*, 16:101–133, 2001. [p13]

S. A. Bustin, V. Benes, J. A. Garson, J. Hellemans, J. Huggett, M. Kubista, R. Mueller, T. Nolan, M. W. Pfaffl, G. L. Shipley, J. Vandesompele, and C. T. Wittwer. The MIQE Guidelines: Minimum Information for Publication of Quantitative Real-Time PCR Experiments. 55(4):611–622. ISSN 0009-9147, 1530-8561. doi: 10.1373/clinchem.2008.112797. URL http://www.clinchem.org/content/55/4/611. [p1]

A. S. Devonshire, R. Sanders, T. M. Wilkes, M. S. Taylor, C. A. Foy, and J. F. Huggett. Application of next generation qPCR and sequencing platforms to mRNA biomarker analysis. *Methods (San Diego, Calif.)*, 59(1):89–100, Jan. 2013. ISSN 1095-9130. doi: 10.1016/j.ymeth.2012.07.021. [p1]

A. J. Durán, M. Pérez, and J. L. Varona. The misfortunes of a trio of mathematicians using computer algebra systems. Can we trust in them? *Notices of the American Mathematical Society*, 2014. ISSN 0002-9920. doi: dx.doi.org/10.1090/noti1173. URL http://www.ams.org/notices/201410/rnoti-p1249.pdf. [p2]

F. C. Fang, R. G. Steen, and A. Casadevall. Misconduct accounts for the majority of retracted scientific publications. *Proceedings of the National Academy of Sciences of the United States of America*, 109(42):17028–17033, Oct. 2012. ISSN 1091-6490. doi: 10.1073/pnas.1212247109. [p2]

C. Gandrud. *Reproducible Research with R and RStudio*. Chapman and Hall/CRC, July 2013. ISBN 9781466572843. [p2]

R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang, and J. Zhang. Bioconductor: open software development for computational biology and bioinformatics. *Genome biology*, 5(10):R80, 2004. ISSN 1465-6914. doi: 10.1186/gb-2004-5-10-r80. PMID: 15461798 PMCID: PMC545600. [p2]

M. Gesmann and D. de Castillo. Using the Google Visualisation API with R. *The R Journal*, 3(2):40–44, Dec. 2011. URL http://journal.r-project.org/archive/2011-2/RJournal_2011-2_Gesmann+de~Castillo.pdf. [p2]

M. Guescini, D. Sisti, M. B. Rocchi, L. Stocchi, and V. Stocchi. A new real-time PCR method to overcome significant quantitative inaccuracy due to slight amplification inhibition. *BMC Bioinformatics*, 9(1):326, July 2008. ISSN 1471-2105. doi: 10.1186/1471-2105-9-326. URL http://www.biomedcentral.com/1471-2105/9/326/abstract. PMID: 18667053. [p4]

H. Hofmann, A. Unwin, and D. Cook. Let Graphics Tell the Story - Datasets in R. *The R Journal*, 5(1):117–130, June 2013. URL http://journal.r-project.org/archive/2013-1/RJournal_2013-1_hofmann-unwin-cook.pdf. [p2]

J. Huggett, J. O'Grady, and S. Bustin. How to make mathematics biology's next and better microscope. *Biomolecular Detection and Quantification*, 2014. ISSN 2214-7535. doi: 10.1016/j.bdq.2014.09.001. URL http://www.sciencedirect.com/science/article/pii/S2214753514000060. [p1]

J. F. Huggett, C. A. Foy, V. Benes, K. Emslie, J. A. Garson, R. Haynes, J. Hellemans, M. Kubista, R. D. Mueller, T. Nolan, M. W. Pfaffl, G. L. Shipley, J. Vandesompele, C. T. Wittwer, and S. A. Bustin. The digital MIQE guidelines: Minimum Information for Publication of Quantitative Digital PCR Experiments. *Clinical Chemistry*, 59(6):892–902, June 2013. ISSN 1530-8561. doi: 10.1373/clinchem.2013.206375. PMID: 23570709. [p1, 11]

D. A. Khodakov and A. V. Ellis. Recent developments in nucleic acid identification using solid-phase enzymatic assays. *Microchimica Acta*, pages 1–14, 2014. ISSN 0026-3672, 1436-5073. doi: 10.1007/s00604-014-1167-z. URL http://link.springer.com/article/10.1007/s00604-014-1167-z. [p1]

T. J. Leeper. Archiving Reproducible Research and Dataverse with R. *The R Journal*, 6(1):NN–NN, 2014. URL http://journal.r-project.org/archive/accepted/leeper.pdf. [p2]

S. Lefever, J. Hellemans, F. Pattyn, D. R. Przybylski, C. Taylor, R. Geurts, A. Untergasser, J. Vandesompele, and RDML consortium. RDML: structured language and reporting guidelines for real-time quantitative PCR data. *Nucleic Acids Research*, 37(7):2065–2069, Apr. 2009. ISSN 1362-4962. doi: 10.1093/nar/gkp056. PMID: 19223324 PMCID: PMC2673419. [p2]

Z. Liu and S. Pounds. An R package that automatically collects and archives details for reproducible computing. *BMC Bioinformatics*, 15(1):138, May 2014. ISSN 1471-2105. doi: 10.1186/1471-2105-15-138. URL http://www.biomedcentral.com/1471-2105/15/138/abstract. [p2]

T. Luo, L. Jiang, W. Sun, G. Fu, J. Mei, and Q. Gao. Multiplex real-time PCR melting curve assay to detect drug-resistant mutations of mycobacterium tuberculosis. *Journal of Clinical Microbiology*, 49 (9):3132–3138, Sept. 2011. ISSN 1098-660X. doi: 10.1128/JCM.02046-10. [p7]

B. D. McCullough and D. A. Heiser. On the accuracy of statistical procedures in microsoft excel 2007. *Computational Statistics & Data Analysis*, 52(10):4570–4578, June 2008. ISSN 0167-9473. doi: 10.1016/j.csda.2008.03.004. URL http://www.sciencedirect.com/science/article/pii/S0167947308001606. [p1, 2]

P. Michna and M. Woods. RNetCDF – A Package for Reading and Writing NetCDF Datasets. *The R Journal*, 5(2):29–37, Dec. 2013. URL http://journal.r-project.org/archive/2013-2/michna-woods.pdf. [p2]

C. A. Milbury, Q. Zhong, J. Lin, M. Williams, J. Olson, D. R. Link, and B. Hutchison. Determining lower limits of detection of digital PCR assays for cancer-related gene mutations. *Biomolecular Detection and Quantification*, 1(1):8–22, Sept. 2014. ISSN 2214-7535. doi: 10.1016/j.bdq.2014.08.001. URL http://www.sciencedirect.com/science/article/pii/S2214753514000047. [p1]

A. A. Morley. Digital PCR: A brief history. *Biomolecular Detection and Quantification*, 1(1):1–2, Sept. 2014. ISSN 2214-7535. doi: 10.1016/j.bdq.2014.06.001. URL http://www.sciencedirect.com/science/article/pii/S2214753514000023. [p1]

P. Murrell. It's Not What You Draw, It's What You Don't Draw. *The R Journal*, 4(2):13–18, Dec. 2012. URL http://journal.r-project.org/archive/2012-2/RJournal_2012-2_Murrell2.pdf. [p2]

J. Oh. Automatic Conversion of Tables to LongForm Dataframes. *The R Journal*, N(2):NN–NN, 2014. URL http://journal.r-project.org/archive/accepted/oh.pdf. [p2]

S. Pabinger, S. Rödiger, A. Kriegner, K. Vierlinger, and A. Weinhäusel. A survey of tools for the analysis of quantitative PCR (qPCR) data. *Biomolecular Detection and Quantification*, 1, 2014. ISSN 2214-7535. [p2, 11]

S. N. Peirson, J. N. Butler, and R. G. Foster. Experimental validation of novel and conventional approaches to quantitative real-time PCR data analysis. *Nucleic Acids Research*, 31(14):e73, July 2003. ISSN 1362-4962. PMID: 12853650 PMCID: PMC167648. [p1]

R Development Core Team. *R Data Import/Export*. R Foundation for Statistical Computing, Vienna, Austria, 2012. URL http://www.R-project.org/. ISBN 3-900051-10-0. [p2, 3]

C. Ritz and A.-N. Spiess. qpcR: an R package for sigmoidal model selection in quantitative real-time polymerase chain reaction analysis. *Bioinformatics*, 24(13):1549–1551, July 2008. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btn227. URL http://bioinformatics.oxfordjournals.org/content/24/13/1549. PMID: 18482995. [p3]

S. Rödiger and M. Burdukiewicz. chipPCR: Toolkit of helper functions to pre-process amplification data, June 2014. URL http://cran.r-project.org/web/packages/chipPCR/index.html. [p3]

S. Rödiger, T. Friedrichsmeier, P. Kapat, and M. Michalke. RKWard: A Comprehensive Graphical User Interface and Integrated Development Environment for Statistical Analysis with R. *Journal of Statistical Software*, 49(9):1–34, 2012. ISSN 1548-7660. URL http://www.jstatsoft.org/v49/i09. [p3, 13]

S. Rödiger, A. Böhm, and I. Schimke. Surface Melting Curve Analysis with R. *The R Journal*, 5(2):37–53, Dec. 2013a. URL http://journal.r-project.org/archive/2013-2/roediger-bohm-schimke.pdf. [p3, 4, 5]

S. Rödiger, P. Schierack, A. Böhm, J. Nitschke, I. Berger, U. Frömmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann, and C. Schröder. A highly versatile microscope imaging technology platform for the multiplex real-time detection of biomolecules and autoimmune antibodies. *Advances in Biochemical Engineering/Biotechnology*, 133:35–74, 2013b. ISSN 0724-6145. doi: 10.1007/10_2011_132. [p1, 11]

S. Rödiger, C. Liebsch, C. Schmidt, W. Lehmann, U. Resch-Genger, U. Schedler, and P. Schierack. Nucleic acid detection based on the use of microbeads: a review. *Microchimica Acta*, pages 1–18, 2014. ISSN 0026-3672, 1436-5073. doi: 10.1007/s00604-014-1243-4. URL http://link.springer.com/article/10.1007/s00604-014-1243-4. [p1, 9]

J. M. Ruijter, M. W. Pfaffl, S. Zhao, A. N. Spiess, G. Boggy, J. Blom, R. G. Rutledge, D. Sisti, A. Lievens, K. De Preter, S. Derveaux, J. Hellemans, and J. Vandesompele. Evaluation of qPCR curve analysis methods for reliable biomarker discovery: bias, resolution, precision, and implications. *Methods (San Diego, Calif.)*, 59(1):32–46, Jan. 2013. ISSN 1095-9130. doi: 10.1016/j.ymeth.2012.08.011. PMID: 22975077. [p1, 3]

J. M. Ruijter, P. Lorenz, J. M. Tuomi, M. Hecker, and M. J. B. v. d. Hoff. Fluorescent-increase kinetics of different fluorescent reporters used for qPCR depend on monitoring chemistry, targeted sequence, type of DNA input and PCR efficiency. *Microchimica Acta*, 181(13-14):1689–1696, Oct. 2014. ISSN 0026-3672, 1436-5073. doi: 10.1007/s00604-013-1155-8. URL http://link.springer.com/article/10.1007/s00604-013-1155-8. [p1]

A.-N. Spiess, C. Feig, and C. Ritz. Highly accurate sigmoidal fitting of real-time PCR data by introducing a parameter for asymmetry. *BMC Bioinformatics*, 9(1):221, Apr. 2008. ISSN 1471-2105. doi: 10.1186/1471-2105-9-221. URL http://www.biomedcentral.com/1471-2105/9/221/abstract. [p3]

P. M. Valero-Mora and R. Ledesma. Graphical User Interfaces for R. *Journal of Statistical Software*, 49(1): 1–8, June 2012. ISSN 1548-7660. URL http://www.jstatsoft.org/v49/i01. [p3]

E. Viturro, C. Altenhofer, B. Zölch, A. Burgmaier, I. Riedmaier, and M. W. Pfaffl. Microfluidic high-throughput reverse-transcription quantitative PCR analysis of liver gene expression in lactating animals. *Microchimica Acta*, 181(13-14):1725–1732, Oct. 2014. ISSN 0026-3672, 1436-5073. doi: 10.1007/s00604-014-1205-x. URL http://link.springer.com/article/10.1007/s00604-014-1205-x. [p1]

M. Zeller, W.-C. L. A. Guazzelli, and G. Williams. PMML: An Open Standard for Sharing Models. *The R Journal*, 1(1):60–65, June 2009. URL http://journal.r-project.org/archive/2009-1/RJournal_2009-1_Guazzelli+et+al.pdf. [p2]

Stefan Rödiger (corresponding author)
Faculty of Natural Sciences
Brandenburg University of Technology Cottbus–Senftenberg
Senftenberg
Germany Stefan.Roediger@hs-lausitz.de

Michał Burdukiewicz
University of Wroclaw
Faculty of Biotechnoloy
Department of Genomics
Wroclaw
Poland michalburdukiewicz@gmail.com

Konstantin Blagodatskikh
All-Russia Research Institute of Agricultural Biotechnology
Center for collective use "Biotechnology"
Moscow
Russia k.blag@yandex.ru

Peter Schierack
Faculty of Natural Sciences
Brandenburg University of Technology Cottbus–Senftenberg
Senftenberg
Germany Peter.Schierack@hs-lausitz.de