BIOINFORMATICS

Vol. 00 no. 00 2014
Pages 1–3

chipPCR: an R Package to Pre-Process Amplification Curve Data

Stefan Rödiger [1]*, Michał Burdukiewicz [2] and PeterSchierack [1]

[1]Faculty of Natural Sciences, Brandenburg University of Technology Cottbus–Senftenberg, Senftenberg, Germany
[2]Department of Genomics, Faculty of Biotechnology, University of Wrocław, Wrocław, Poland

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: The quantitative real-time polymerase chain reaction (qPCR) and quantitative isothermal amplification (qIA) are standard methods for quantification of nucleic acids. Numerous real-time read-out technologies with different technical foundation have been developed. Despite the continuous interest in amplification based techniques, there are only few tools for amplification data pre-processing. Especially, during development of new instruments a transparent tool for precise control of raw data is indispensable.

Results: *chipPCR* is an **R** package for pre-processing and quality analysis of amplification curve data. The package takes advantage of **R**'s *S4* object model and offers an extensible environment. Functions to simulate, normalize, baseline, impute missing values, to smooth amplification curves and a function to to detect the start and end of an amplification curve are part of the software. We implemented a 5-point stencil for derivative interpolation, which is unique in **R**. Statistical tests, plots for data quality management, amplification efficiency/quantification cycle calculation, and 22 data sets from various qPCR and qIA experiments are part of the package. The structure of *chipPCR* was designed for integration as GUI's (web-based and standalone *shiny* applications) and for report generation.

Availability: Stabel: http://cran.r-project.org/web/packages/chipPCR
Source code: https://github.com/michbur/chipPCR

Contact: stefan.roediger@hs-lausitz.de

Supplementary: Supplementary data are available at Bioinformatics online.

1 INTRODUCTION

Quantitative polymerase chain reaction (qPCR) and quantitative isothermal amplification (qIA) are standard methods to amplify nucleic acids. These methods are used in real-time monitoring technologies, such as our previously reported VideoScan technology, microfluidic systems, point-of-care devices, and qPCR cyclers. Real-time technologies enable the quantification of nucleic acids by calculation of specific curve parameters like the quantification point (Cq) and the amplification efficiency (AE) (Rödiger *et al.*, 2013a, 2014; Pabinger *et al.*, 2014). Fundamental steps of amplification curve analysis are: (1) raw data read-in, (2) pre-processing (e.g., noise reduction), (3) amplification curve processing (e.g., Cq calculation), (4) post-processing and quantification, (5) data export/report generation. **R** belongs to the most used bioinformatic s tools and is a rapid adopter for various technologies like digital PCR, NanoString nCounter Platform, and qPCR (Waggott *et al.*, 2012; Pabinger *et al.*, 2014). Most qPCR **R** packages focus on the read-in and (post)-processing of data from commercial qPCR systems. **R** packages for the steps 1. and 3.–5. are available (Pabinger *et al.*, 2014; Perkins *et al.*, 2012; McCall *et al.*, 2014; Gehlenborg *et al.*, 2013).

However, there is no **R** package for pre-processing and quality analysis of amplification curve data. Pre-processing is in most commercial cyclers a "black box", which sets serve limits to reproducible research (Leeper, 2014). Developmental qPCR and qIA technologies depend on tools to pre-process the raw data. Pre-processing algorithms remove stochastic errors and artefacts (Fig. **??**). Pre-processing addresses raw data inspection, raw data transformation in a format for successive analysis steps (e.g., smoothing, imputation), data reduction (e.g., removal of invalid sets) and data quality management. The data quality of experimental instruments is often not ready for end-user analysis but it is important to use as many raw data as possible. Misinterpretations are more likely if "arbitrary" corrections are performed. A manual alteration is in contradiction to reproducible research. The *chipPCR* package ("Lab-on-a-**Chip**" & **PCR**) was developed to fill this gab and to automatize pre-processing, data analysis/visualization and to offer a quality control for the statistical data analysis of qPCR and qIA experiments. **R** is very powerful to reproduce analysis on different platforms, to adopt to changing experimental setups and offers sophisticated statistical tools. Moreover, it is desirable to set up workflows in an open environment, which enables downstream analyses and which offers powerful tools for data visualizations and automatic report generation. The target audience encompasses developers and users who process raw data of commercial systems.

2 IMPLEMENTATION

chipPCR package was implemented in the **R** software environment as described elsewhere (R Core Team, 2014; Rödiger *et al.*, 2012). *chipPCR* is a relative of the *MBmca* (Rödiger *et al.*, 2013b), the *RDML* (Blagodatskikh *et al.*, 2014), and the *dpcR* (Pabinger *et al.*, 2014) packages but focusses on pre-processing of amplification curves. The package functions are listed in Suppl. Sect. **??**. The supplemental material (packages vignette), uses Donald

*to whom correspondence should be addressed

Knuth's literate programming principle (Knuth, 1984) to present the source code in a convenient way. *chipPCR*'s naming convention is *period.separated* (Bååth, 2012). We use **R**'s object model *S4* class system (see *methods* package) to separate between interface and implementation. Other than in **R**'s *S3* class system, needs *S4* to declare classes, slots, methods relationships explicitly and to establish formal declarations of methods. Therefore, the number and types of slots in an instance of a class have to be established at the time the class definition and the objects from the class are validated against this definition and have to comply to it at any time. *S4* methods are declared by calls to *setMethod* together with the names and signatures of the arguments. Signatures are used for identification of classes of one or more arguments of the methods. *setGeneric* can be used to declare generic functions. *S4* classes require a higher effort than *S3* classes, but assures strictly objects in a class have the required slots, that data in the slots have consistent names and classes, and enable to include additional information (e.g., results, parameters). As perquisite for high-throughput technologies we avoided loops in the core structures and use partially parallel computing (*smoother* function) to keep the code fast. *chipPCR* includes a set of classes for plotting. The output of our custom made plots is minimalistic, but many parameters can be adjusted directly or by the ellipse parameter.

Graphical user interface (GUI) are important to spread software. **R** offers several GUI projects to chose from (Rödiger *et al.*, 2012). Recently, the *shiny* (RStudio and Inc., 2014) framework to build and deploy GUI's for the desktop (web browser) or services for interactive web applications emerged. *shiny* enables to build plugin-like applications with highly customizable widgets (e.g., sliders, plots, reports) for a efficient extension. *shiny* applications update live and interactively. The user interfaces can be built entirely using **R** and operates in any **R** environment (cross-platform). Currently, the functions *AmpSim*, *th.cyc*, *bg.max* and *amptester* are part of *shiny* GUIs. It is possible to run the applications as service on a server (e.g., http://michbur.shinyapps.io/MFIaggr_gui/), on the local desktop (e.g., Fig. **??**), or as deployed from an external source for a local **R** installation. For further details refer to (Suppl. Section **??**). *shiny* in combination with **R** is a foundation to build monolithic systems to parse, pre-process and analyze amplification curve data in a combined work-flow. Some functionality of *chipPCR* is used in experimental plugins for the **RKWard** GUI (Pabinger *et al.*, 2014).

We have chosen not to rely on specialized parser but use native **R** workspaces, and dedicated **R** packages as default data format for import and export as described elsewhere . *chipPCR* presents *S4* objects with tailored summary and plot methods. Data sets are an essential element of reproducible research (Leeper, 2014). Our package contains 22 data sets from commercial and experimental cyclers along with the experimental settings (e.g., helicase dependent amplification (HDA)) (Suppl. Sect. **??**).

3 EXAMPLE: QUALITY ANALYSIS

MFIaggr is a powerful analytical and graphical tool for fast multiple comparison of the cycle dependent signal dispersion and distribution (Fig. 1). The continuous response variable y_i' is used to describe the relationships to one or more continuous predictor variables $y_1, ..., y_n$. Use cases include the comparison of independent reaction vessels or the analysis of replicate experiments (Suppl. Sect. **??**). In particular, this function might be useful for quality management during the development of high-throughput technologies. An analysis via a the *shiny MFIaggr.gui* app is shown in Fig. **??**.

4 RESULTS AND CONCLUSIONS

There is an ongoing development for qPCR and qIA technologies. *chipPCR* is the first **R** package for the pre-processing and raw data quality analysis of amplification curve data of such systems. Though, *chipPCR* primarily targets pre-processing we also implemented standard methods to process amplification curve data. Functions of *chipPCR* are embeddable in customized routines with other packages (see Suppl.). For example, the packages *dpcR* and *MBmca* depend on *chipPCR* technology. We showed that *chipPCR* is build from smaller blocks and show how users can do estimation of background by hand, solely by *inder*, *smoother* (*smoother* will be a method of smoothing in *inder*) and by putting data in *bg* object with *summary-der* for SDm and SDM. It should be quite easy even for an inexperienced user. We claim that the modular structure of *chipPCR* package allows user to perform flexible data analysis adjusted to their needs. A limitation of all **R** packages related to qPCR and qIA is the lack of a comprehensive GUI, which we filled with our *shiny* GUI approach.

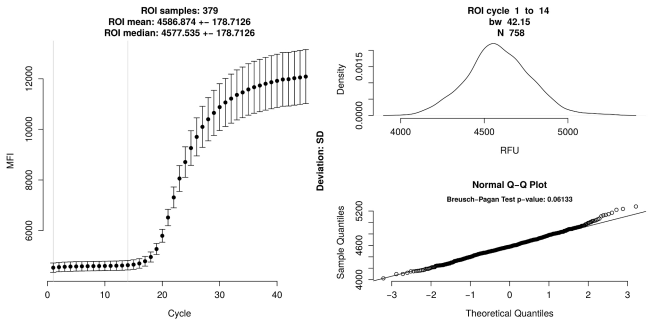

Fig. 1. Amplification curve analysis of 379 replicates. Cycles 1 to 14 was selected as region of interest (ROI) for *MFIaggr* to analyzes the cycle-dependent variance (left panel) and gives a density plot (right upper panel) and quantile-quantile analysis (right lower panel). The plots indicates that the data of the background range are normal distributed.

ACKNOWLEDGEMENT

Grateful thanks belong to the **R** community.

Funding: This work was funded by the BMBF InnoProfile–Projekt 03 IPT 611X.

Conflict of Interest: none declared.

REFERENCES

Bååth, R. (2012). The State of Naming Conventions in R. *The R Journal*, **4**(2), 74–75.

Blagodatskikh, K. A., Roediger, S., and Burdukiewicz, M. (2014). Importing real-time thermo cycler (qPCR) data from RDML format files.

Gehlenborg, N., Noble, M. S., Getz, G., Chin, L., and Park, P. J. (2013). Nozzle: a report generation toolkit for data analysis pipelines. *Bioinformatics*, **29**(8), 1089–1091.

Knuth, D. E. (1984). Literate Programming. *The Computer Journal*, **27**(2), 97–111.

Leeper, T. J. (2014). Archiving Reproducible Research and Dataverse with R. *The R Journal*, **6**(1), NN–NN.

McCall, M. N., McMurray, H. R., Land, H., and Almudevar, A. (2014). On non-detects in qPCR data. *Bioinformatics (Oxford, England)*.

Pabinger, S., Rödiger, S., Kriegner, A., Vierlinger, K., and Weinhäusel, A. (2014). A survey of tools for the analysis of quantitative PCR (qPCR) data. *Biomolecular Detection and Quantification*.

Perkins, J. R., Dawes, J. M., McMahon, S. B., Bennett, D. L., Orengo, C., and Kohl, M. (2012). ReadqPCR and NormqPCR: R packages for the reading, quality checking and normalisation of RT-qPCR quantification cycle (Cq) data. *BMC Genomics*, **13**(1), 296. PMID: 22748112.

R Core Team (2014). R: A Language and Environment for Statistical Computing.

Rödiger, S., Friedrichsmeier, T., Kapat, P., and Michalke, M. (2012). RKWard: A Comprehensive Graphical User Interface and Integrated Development Environment for Statistical Analysis with R. *Journal of Statistical Software*, **49**(9), 1–34.

Rödiger, S., Schierack, P., Böhm, A., Nitschke, J., Berger, I., Frömmel, U., Schmidt, C., Ruhland, M., Schimke, I., Roggenbuck, D., Lehmann, W., and Schröder, C. (2013a). 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.

Rödiger, S., Böhm, A., and Schimke, I. (2013b). Surface Melting Curve Analysis with R. *The R Journal*, **5**(2), 37–53.

Rödiger, S., Liebsch, C., Schmidt, C., Lehmann, W., Resch-Genger, U., Schedler, U., and Schierack, P. (2014). Nucleic acid detection based on the use of microbeads: a review. *Microchimica Acta*, **181**(11-12), 1151–1168.

RStudio and Inc. (2014). shiny: Web Application Framework for R. R package version 0.10.1.

Waggott, D., Chu, K., Yin, S., Wouters, B. G., Liu, F.-F., and Boutros, P. C. (2012). NanoStringNorm: An Extensible R Package For the Pre-Processing of NanoString mRNA and miRNA Data. *Bioinformatics*, page bts188.