Surface Melting Curve Analysis with R

by Stefan Rödiger, Alexander Böhm and Ingolf Schimke

Abstract Nucleic acid *Melting Curve Analysis* is a powerful method to investigate the interaction of double stranded nucleic acids. Many researchers rely on closed source software which is not ubiquitously available, and gives only little control over the computation and data presentation. R in contrast, is open source, highly adaptable and provides numerous utilities for data import, sophisticated statistical analysis and presentation in publication quality. This article covers methods, implemented in the **MBmca** package, for DNA Melting Curve Analysis on microbead surfaces. Particularly, the use of the second derivative melting peaks is suggested as an additional parameter to characterize the melting behavior of DNA duplexes. Examples of microbead surface Melting Curve Analysis on fragments of human genes are presented.

Introduction

Melting Curve Analysis

Nucleic acid *Melting Curve Analysis* (MCA) is a central step in nucleic acid[1] interaction studies, identification of specific DNA sequences after quantitative real-time PCR, genotyping or detection of *Single Nucleotide Polymorphisms*[2] (SNP) both in solution and on surfaces (Ririe et al., 1997; Gundry et al., 2003; Sekar et al., 2005; Zhou et al., 2005; Rödiger et al., 2012b). A review of the literature revealed that there is an ongoing demand for new bioanalytical devices to perform MCAs. This includes lab-on-chip systems or the recently published *VideoScan* platform (Rödiger et al., 2012b, see following section). Some of these systems offer software solutions for MCA but often custom made software is required. Although bioanalytical devices break new ground to meet criteria like high multiplex levels or new detection-probe-systems the fundamental concept of MCA remains unchanged.

The MCA is a real-time monitoring of a heat-induced double stranded nucleic acid dissociation which can be monitored by the change of the referenced mean/median fluorescence intensity (MFI)[3] at a defined temperature (Figure 1).

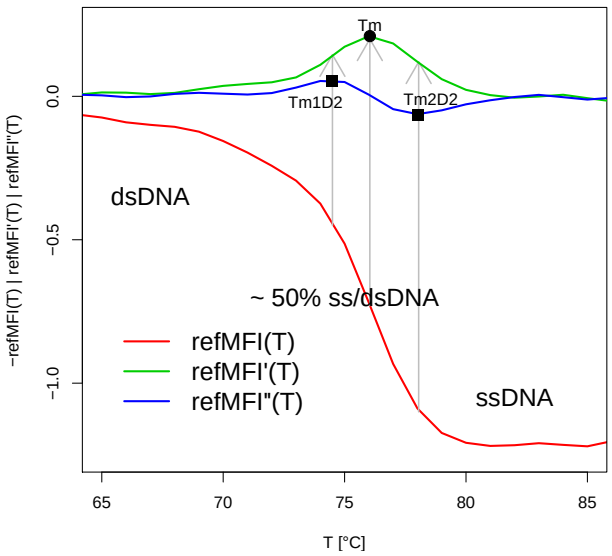

Figure 1: Melting curve and melting peaks of double stranded DNA. Raw data curves ($refMFI(T)$ —), approximate first derivative ($refMFI'(T)$ —) and second derivative ($refMFI"(T)$ —). Peak values: Tm is the maximum of the first derivative. Tm_1^{D2} and Tm_2^{D2} are the extremes (a minimum and a maximum) of the second derivative. dsDNA, double stranded DNA; ssDNA, single stranded DNA.

[1] Nucleic acids herein refer to deoxyribonucleic acids (DNA) and ribonucleic acids (RNA).

[2] *Single Nucleotide Polymorphisms* (SNP) are caused by exchanges of single nucleotides.

[3] In the past absorbency was the quantitative measure. New devices measure fluorescence intensities mediated by nucleic acid-intercalating fluorophores (e. g., *EvaGreen*®) or fluorophore labeled DNA probes (e. g., *Molecular Beacons*, Gašparič et al., 2010; Rödiger et al., 2012a). The values are often reported as *RFU* (Relative Fluorescence Units), MFI (Mean/Median Fluorescence Intensity) or $refMFI$ (Referenced Mean/Median Fluorescence Intensity).

By definition, the *melting point* (Tm) is the inflection point of the melting curve. On molecular level circa 50% of the nucleic acids are dissociated at Tm. The *melting peak* (Equation 1) can be determined from the first negative derivative (Equation 2) of the melting curve. At this temperature peak the rate of change is maximal. The Tm is highly reproducible, thus can be used as a "characteristic identity" to distinguish nucleic acid species.

$$Tm = \max(refMFI'(T)) \tag{1}$$

$$refMFI'(T) = -\frac{d(refMFI)}{d(T)} \tag{2}$$

To the best of our knowledge we are the first to suggest the peak values, designated Tm_1^{D2} and Tm_2^{D2} (Equation 3 and 4), of the second derivative (Equation 5) as additional measures for the characterization of nucleic acid melting processes on microbead surfaces. They show the maximal rate of change of $refMFI'(T)$. The corresponding temperature values (abscissa) at the inflection points of $refMFI'(T)$ occur where $refMFI"(T)$ reaches a (local) minimum and a (local) maximum, respectively (Figure 1). Both values offer additional quantitative measures to describe early and late phases of the melting process. The Tm_1^{D2} quantifies the maximal acceleration and Tm_2^{D2} the maximal deceleration of the dissociation process. The deceleration starts in the middle of the process, at Tm.

$$Tm_1^{D2} = \max(refMFI"(T)) \tag{3}$$

$$Tm_2^{D2} = \min(refMFI"(T)) \tag{4}$$

$$refMFI"(T) = -\frac{d^2(refMFI)}{d(T)^2} \tag{5}$$

Surface Melting Curve Analysis

In Rödiger et al. (2012b) we reported the *VideoScan* platform which provides a technology for various bioanalytical applications[4] with the focus on highly multiplex kinetic analysis. The *VideoScan* platform consists of a fully automated fluorescence microscope, a modular software package for data acquisition and a heating/cooling-unit (HCU)[5] for micro volume ($\leq 20\ \mu L$) samples. We developed temperature controlled assays to detect and analyze nucleic acid probes on the surface of microbeads.

In brief, different thermo-tolerant microbead populations, defined by varying ratios of two impregnated fluorophores, are included in the reaction. Each microbead population presents gene specific capture probes on their surfaces (Figure 2).

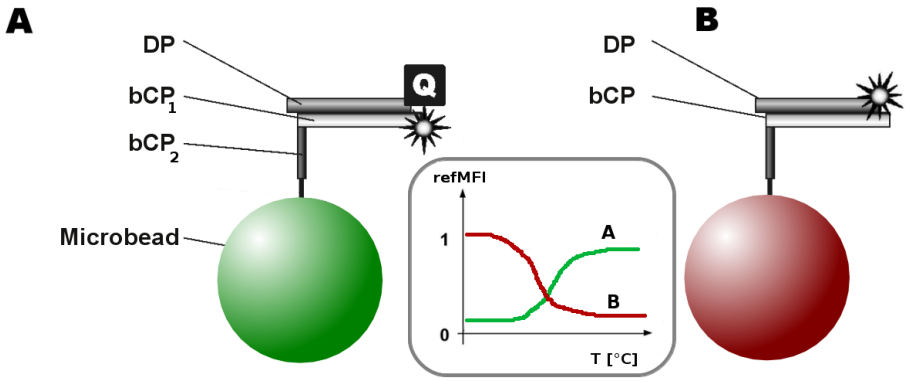

Figure 2: Principles of microbead *direct hybridization* probe systems. A) The *FRET assay* uses microbead bound capture probe (*bCP*) with the fluorophore- (*) and quencher-labeled (**Q**) detection probes (*DP*). A *bCP* consists of different regions (e. g., bCP_1, bCP_2) ready to hybridize with a complementary *DP*. B) The *standard assay* uses non-labeled *bCP*s which hybridize with fluorophore-labeled *DP*s. Inset) Both probe systems differ in the curve shape resulting from the melting process.

[4]These include autoimmune cell pattern recognition, microbead-based assay for DNA and proteins. For details see Willitzki et al. (2012) and Rödiger et al. (2012b).

[5]The heating/cooling-unit (HCU) is based on peltier elements and has a performance similar to conventional thermal cyclers (e. g., iQ5 (Bio-Rad Laboratories)). For details see Rödiger et al. (2012b).

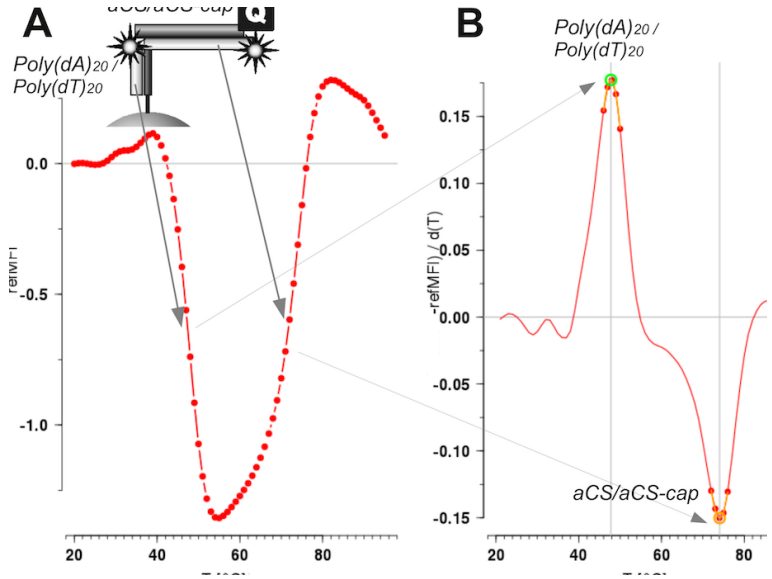

Figure 3: Melting curve and melting peaks. The data were obtained from a MCA experiment on microbead surfaces. Top left inset: Two DNA detection probes ($Poly(dA)_{20}$, *aCS*), were hybridized to a complementary microbead bound capture probe. A) While applying a temperature gradient (20 °C to 95 °C, 1 °C per step) the change of the $refMFI$ is monitored. B) This probe system exhibits two melting peaks. Analysis showed the first (positive) peak for $Poly(dA)_{20}$ (~48 °C) followed by a second (negative) peak of *aCS* (~74 °C).

Particularly, short 3′ or 5′ quencher/fluorophore-labeled probes were used[6]. The capture probes (*bCP*) are complementary to detection probe (*DP*) in solution (Figure 2 and inset Figure 13). Signals are mediated by temperature dependent dehybridization events between *bCP*s and *DP*s. All probe systems described here and our related works are characterized by one or two significant melting peaks at different temperatures and/or different signs (Figure 3). Due to different fluorophore / quencher combinations and probe systems (e. g., direct hybridization, dual-hybridization probes) the sign of a Tm peak value was designed to be either positive or negative (see Figure 3B). Transferred to applications in screening processes all subsequent analysis would be reduced to the identification of distinct Tms and their intensity of the corresponding microbead populations.

Why use R for Melting Curve Analysis?

In biomedical research many scientists rely on closed source software which gives only little control over the computation and data presentation. Most importantly reproduction of data and calculations from other research is limited or impossible. Just recently this was discussed in the two journals *Nature* and *Science* (Ince et al., 2012; Morin et al., 2012). Closed software tied to limited tasks, hinders the import of custom data and gives no control over or insight into the source code and therefore is not ideal for research. Basically any open computing language or tool can be used to overcome these issues. But R fulfills all requirements mentioned above. It is in an open system with numerous utilities for data import, processing, sophisticated statistical analysis and presentation. Many R packages are peer-reviewed and undergo an intensive testing. Most importantly R is open source and therefore methods or results can be reproduced independently.

One implementation for MCA with R is available from the excellent **qpcR** package by Ritz and Spiess (2008). The `meltcurve()` function provides a sophisticated method to process melting curve data from qPCR experiments in solution. It uses automatic optimization procedures to smooth and fit curves. This function is ideal for the identification of multiple Tms, the calculation of the peak area and high resolution data.

[6]The DNA probe sequences, microbeads, probe immobilization process and experimental set up were described in Rödiger et al. (2011, 2012b). The fluorescence dye *Atto 647N* and the quencher *BHQ2* were used generally. Intercalating dyes were not used since they are known to alter the melting process (Gudnason et al., 2007).

Motivation for the MBmca package

There is an ongoing interest to understand the melting behavior of nucleic acids on surfaces. Particularly the effects of the reaction environment, e. g., the microbead surface change due to functional groups or the density of *bCP* has not been investigated intensively for multiplex microbead assays. During the development of the *VideoScan* platform a simple and lightweight method for automatic screening, quality control and automatic detection of a limited number of melting peaks was needed. This article describes the **MBmca**[7] (Roediger, 2013) package which was developed as an approach for statistical MCA on microbeads. The ambition of the **MBmca** is not to compete with the **qpcR** or other R packages but rather to extend the scope of R for MCA on surfaces.

Implementation of MCA in the MBmca package

Functions and data sets of MBmca

The **MBmca** package includes the functions `MFIerror()`, `mcaPeaks()`[8] and `mcaSmoother()` for data inspection and preprocessing and `diffQ()`, `diffQ2()` for MCA. The data sets `DualHyb`, `DMP` and `MultiMelt` are raw fluorescence data measured with the *VideoScan* platform (Rödiger et al., 2012b) on microbead surfaces. The data are arranged as `data.frames` starting in the first column with the temperature (°C) followed by the fluorescence values (refMFI). The package has a dependency to **robustbase** (Rousseeuw et al., 2013) and uses mainly standard R functions. Elementary steps, e. g., data import, are similar to other R functions and thus used as described elsewhere (Venables et al., 2013).

Inspection of raw fluorescence data

One of the fundamental strengths of the MCA on microbead surfaces is the achievable multiplex level. The `MFIerror()` function was developed for a fast multiple comparison of the temperature dependent variance of $refMFI$. `MFIerror()` returns an object of the class `data.frame` with columns "Temperature", "Location" (Mean, Median), "Deviation" (Standard Deviation, Median Absolute Deviation) and "Coefficient of Variation".

- The argument `errplot` (default) sets `MFIerror()` to plot the results. In the default setting (`CV = FALSE`) the mean with the standard deviations is plotted. Using the argument `rob = TRUE` the median and the median absolute deviation (MAD) are plotted instead of the mean and standard deviation.
- If `CV` is true the coefficient of variation (CV) is plotted. Setting the argument `RSD = TRUE` shows the relative standard deviation (RSD) in percent.

In an example the mean raw fluorescence from multiplex melting curves of twelve microbead populations was evaluated for the probes *HPRT1* and $MLC-2v$ (`MultiMelt` data set). The probe system used corresponds to Figure 2A. Ideally the variance between the twelve microbead populations is low. `MFIerror()` takes the first column of `MultiMelt` as temperature value and columns 2 to 13 for the probes *HPRT1* and columns 14 to 25 for $MLC-2v$, respectively.

```
# Load MultiMelt data set.
data(MultiMelt)

# MFIerror for the HRPT1 data (column 2 to 13).
# The default settings of MFIerror show the the mean fluorescence and the
# standard deviation at a defined temperature.
MFIerror(MultiMelt[, 1], MultiMelt[, 2:13])

# MFIerror on the MLC-2v data (column 14 to 25).
MFIerror(MultiMelt[, 1], MultiMelt[, 14:25])
```

The corresponding plots are shown in Figure 4. The curves indicate that the different microbead populations show similar melting curve shapes but differ in height. At higher temperatures the values vary.

[7] **M**icro**B**ead **m**elting **c**urve **a**nalysis.

[8] `mcaPeaks()` is a function which can be used to estimate the number and location of the approximate local minima and maxima of melting curve data. This can be used to define a temperature range for MCA, melting curve quality control or peak height threshold definition (see Roediger 2013).

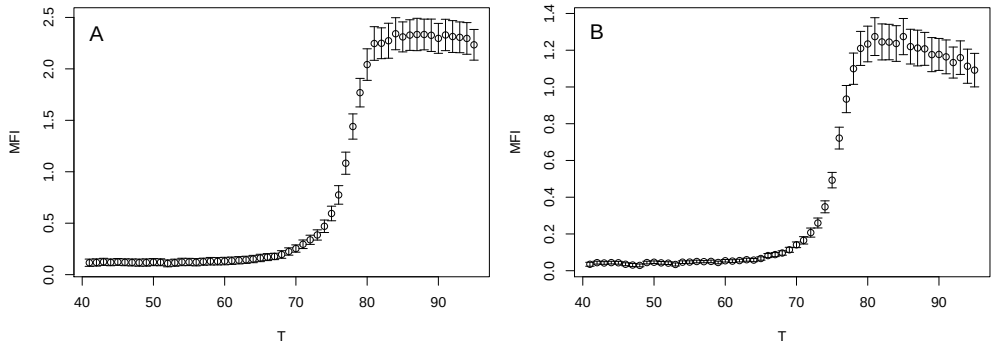

Figure 4: Inspection of the mean raw fluorescence from multiplex melting curves. `MFIerror()` with the argument `rob = FALSE` was used to compare the mean and the standard deviationof the temperature dependent fluorescence on twelve microbead populations for A) *HPRT1* and B) $MLC-2v$.

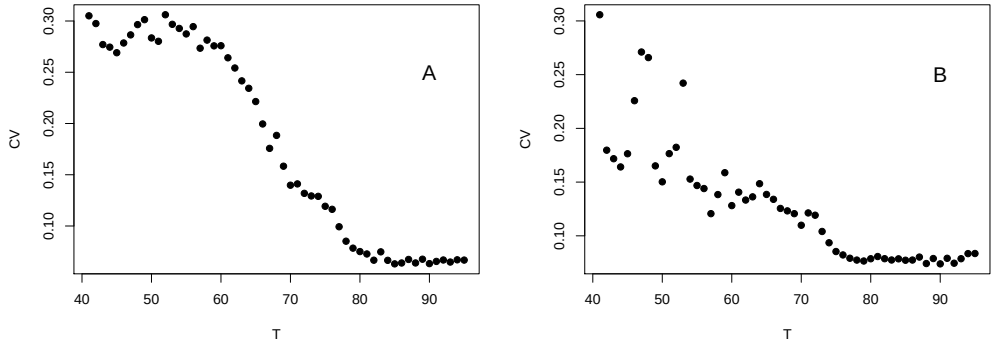

Figure 5: Comparison of the absolute coefficient of variation from multiplex melting curves. `MFIerror()` with argument `CV = TRUE` was used to plot the absolute coefficient of variation for twelve microbead populations with A) *HPRT1* or B) $MLC-2v$.

When `MFIerror()` is used with the argument `CV = TRUE` the coefficient of variation is presented (Figure 5). In the example all CV values are low (< 0.3) and even decrease with increasing temperatures for both *HPRT1* and $MLC-2v$.

We questioned how a single microbead population differs from the average of all microbead populations. The mean output of `MFIerror()`, with the argument `errplot = FALSE`, was subtracted by the fluorescence of *HPRT1* or $MLC-2v$. The results were assigned to `HPRT1.mean` and `MLC2v.mean`.

```
# Load MultiMelt data set.
data(MultiMelt)

# Use MFIerror to calculate the mean fluorescence for HRPT1 and MLC-2v over all
# twelve microbead populations.
HPRT1.mean <- MFIerror(MultiMelt[, 1], MultiMelt[, 2:13], errplot = FALSE)
MLC2v.mean <- MFIerror(MultiMelt[, 1], MultiMelt[, 14:25], errplot = FALSE)

# Draw figures on the graphics device in a 2x6 array
par(mfrow = c(2, 6))

# Calculate the difference between the fluorescence of a single microbead population
# and the average of all twelve microbead populations. Plot the results.
for (i in 1:12) {
  tmp.HPRT1 <- MultiMelt[, i + 1] - HPRT1.mean[, 2]
  tmp.MLC2v <- MultiMelt[, i + 13] - MLC2v.mean[, 2]
  plot(MultiMelt[, 1], tmp.HPRT1, main = paste("Pop", i, sep = ": "),
       pch = 19, ylim = c(-0.28, 0.28), xlab = "T", ylab = "delta")
  abline(h = 0, col = "black")
  abline(v = 65, col = "blue")
  points(MultiMelt[, 1], tmp.MLC2v, pch = 15, col = 2)
}
```

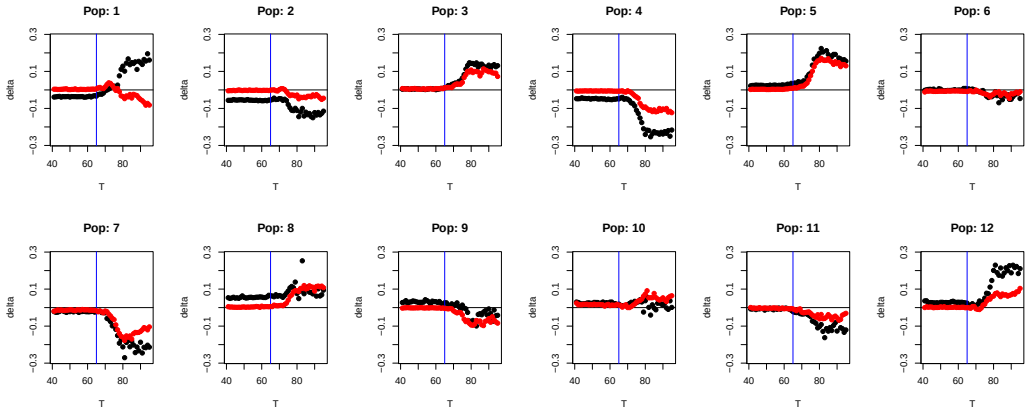

Figure 6: Difference plot of the fluorescence values. The temperature dependent mean fluorescence of twelve microbead populations, determined with `MFIerror()`, was subtracted from the fluorescence of single population ("Pop: "). • $HPRT1$, • $MLC-2v$, — base line, — start of the melt process.

Figure 6 shows low differences (delta) at temperatures below 65 °C due to near complete quenching. Above this temperature (start of the DNA probe strand dissociation) the differences to the mean fluorescence of all microbead populations grow. Apart from "Pop: 1" changes appear systematically which indicates that the *bCP/DP*s have a similar melting behavior on all microbead populations.

Preprocessing of raw fluorescence data

The differentiation is the central step of the MCA. Accessible textbook information confirms that differentiation may result in the amplification of noise. To reduce the noise, R provides numerous possibilities to fit smooth functions and use filter functions. Such operations may alter the curve shape considerably and thus lead to artificial results. Smooth functions available in R include the moving average (`filter()`, **stats**), the LOWESS smoother (`lowess()`, **stats**) which applies locally-weighted polynomial regression, fitting of a local polynomial regression (`loess()`, **stats**), Savitsky-Golay filter (`sgolayfilt()`, **signal**; The signal Developers, 2013), cubic splines (`smooth.spline()`, **stats**) or Friedman's SuperSmoother (`supsmu()`, **stats**). Although smoothed data might provide the impression of high quality data no guarantee for optimal results or that no peaks were artificially introduced is given. A good practice is to visualize the output combined with the original data. `mcaSmoother()` uses `smooth.spline()` and contains further helper function. `mcaSmoother()` should be used if the data may contain missing values, high noise or if the temperature resolution of the melting curve data is low ($\geq$ 0.5 °C / step) in order to correct the problems automatically.

- Measurements from experimental systems may occasionally include missing values (`NA`). Function `mcaSmoother()` uses `approx()`[9] to fill up `NA`s under the assumption that all measurements were equidistant. The original data remain unchanged and only the `NA`s are substituted.
- `mcaSmoother()` calls `smooth.spline()` to smooth the curve. Different strengths can be set using the argument `df.fact` (default 0.95). Internally it takes the degree of freedom value from the spline and multiplies it with a factor between 0.6 and 1.1. Values lower than 1 result in more strongly smoothed curves.
- If the argument `bgadj` is set `TRUE`, `bg` must be used to define a temperature range for a linear background correction[10]. The linear trend is estimated by a robust linear regression using `lmrob()`. In case criteria for a robust linear regression are violated `lm()` is used automatically.
- The argument `Trange` can be used to define a temperature range of the analysis.
- To scale the fluorescence a *Min-Max normalization* (Equation 6) between 0 and 1 can be used by setting the argument `minmax` to `TRUE`. This is useful if the fluorescence values between samples vary considerably, for example due to high background. An advantage of this normalization is the preservation of the relationships between the values. However, on surfaces normalization

[9] Besides `approx()` (**stats**) further functions are available from the **zoo** package (Zeileis and Grothendieck, 2005, e. g., `na.approx()`, `na.spline()`) and the **delftfews** package (Frasca, 2012, `na.fill()`, `na.interpolate()`). `approx()` was integrated since further dependencies are omitted and a linear interpolation is used.

[10] Some software packages include automatic linear or non-linear background correction which work not reliable in many cases. The examples in Figure 4 use identical probe systems (direct hybridization) but different *bCP/DP* combinations. In theory the melting curve shape should be very similar. Particularly values at temperatures above 80 °C vary strongly. Therefore only a simple linear background correction was implemented.

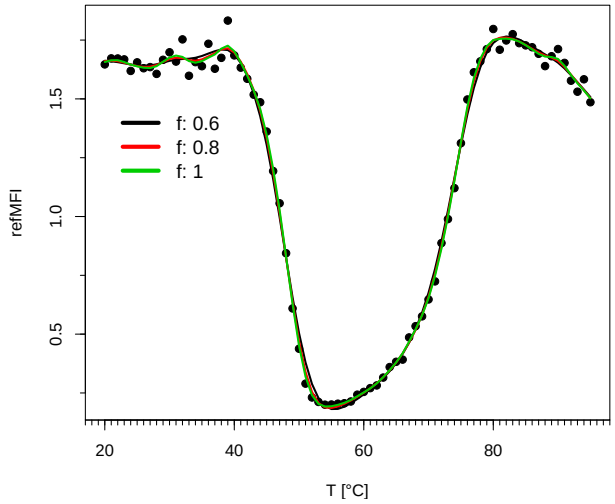

Figure 7: Raw fluorescence values (•) versus the temperature. Smoothed curves (f: 0.6 —, f: 0.8 —, f: 1 —) using `mcaSmoother()` with different strengths (f) to smooth the curves.

should be used with caution because it might lead to the false impression that all microbeads carried equal quantities of *bCP*.

- The argument `n` uses the `spline()` function to increase the temperature resolution of the melting curve data by n-times the length of the input temperature (see `mcaSmoother()` examples in Roediger 2013).

$$refMFInorm = \frac{refMFI - \min(refMFI)}{\max(refMFI) - \min(refMFI)} \tag{6}$$

`mcaSmoother()` returns an object of the class "`data.frame`" with the columns "x" (temperature) and "y" (fluorescence values). These can be used to plot the preprocessed data. For example, three arbitrary chosen strengths to smooth the curves (f: 0.6, 0.8, 1.0) were tested. Data from the `DMP` data set were used as follows:

```
# Load DMP data set.
data(DMP)

# Create plot with raw data.
plot(DMP[, 1], DMP[, 6], xlim = c(20, 95), xlab = "T [C]",
     ylab = "refMFI", pch = 19, col = 8)

# Add minor tick marks to the abscissa.
require(Hmisc); minor.tick(nx = 20)
```

The function `mcaSmoother()` is used in a loop to smooth the curve with user defined strengths (f). Wrapped into `lines()` it draws[11] the corresponding curves directly (Figure 7). In comparison to the original data a low filter strength (f: 1) is sufficient.

```
# Define three filter strengths (highest (0.6) to lowest (1.0)) and assign them
# to df.fact. Smooth the raw data and add the results as lines to the plot.
f <- c(0.6, 0.8, 1.0)
for (i in 1:3) {
  lines(mcaSmoother(DMP[, 1], DMP[, 6], df.fact = f[i]), col = i, lwd = 2)
}
# Add a legend to the plot with the filter strengths.
legend(20, 1.5, paste("f", f, sep = ": "), cex = 1.2, col =  1:3,
       bty = "n", lty = 1, lwd = 4)
```

In the next example `mcaSmoother()` was used with different arguments to (i) smooth the data, (ii) remove the background and (iii) to perform a Min-Max normalization. Data for *HPRT1* (Figure 8A) were taken from the `MultiMelt` data set. The plot of Figure 8B implies that it is sustainable to smooth

[11] A fine grained abscissa was created with `minor.tick()` from the **Hmisc** package (Harrell Jr et al., 2013) to enhance the plot.

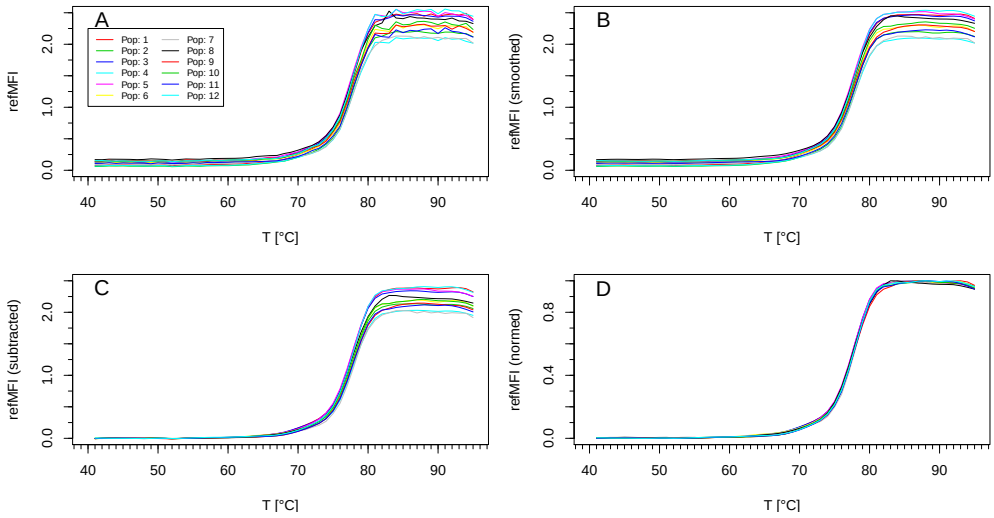

Figure 8: Preprocessing of *HPRT1* with `mcaSmoother()`. A) Raw fluorescence data of melting curves from 12 microbead populations ("Pop: "). B) Smoothed curves with the default settings (f: 0.95). C) The smoothed curves after background reduction (`bg = c(41, 61)`, i.e., 41 °C to 61 °C) and linear trend correction. D) Min-Max normalized curves.

the curve. Not immediately obvious, the twelve microbead populations have different final signal intensities. This is due to the different quantities of surface bound *bCP*s (not shown). However, this is easily visualized after a background correction (Figure 8C). All curves appear similar in shape after the Min-Max normalization. This indicates that there are no substantial differences between the microbead populations which obfuscates further MCAs (Figure 8D).

Calculation of the melting peaks

The functions `diffQ()` and `diffQ2()` are used to calculate Tm, Tm_1^{D2} and Tm_2^{D2} (Figure 1) and to perform simple graphical operations (e. g., show derivatives). Basically, both functions do not require smoothed data[12] for the MCA. However, it is recommended to use `mcaSmoother()` as a starter function to preprocess (e.g., moderate smoothing, missing value removal, type check) the data automatically. First the approximate Tm, Tm_1^{D2} and Tm_2^{D2} are determined as the `min()` and/or `max()` from the derivatives[13] according to Equation 1, 3 and 4. This approximate peak value is the starting-point for an accurate calculation. The function takes a defined number n (maximum 8) of the left and the right neighbor values and fits a quadratic polynomial[14]. The quadratic regression `lm(Y ~ X I(X^2))` of the X (temperature) against the Y (fluorescence) range gives the coefficients. The optimal quadratic polynomial is chosen based on the highest adjusted R-squared value ($R^2_{adj.}$). In the example of Figure 10 two left and right neighbors were required. The coefficients are used to calculate the root of the quadratic polynomial and thus to determine Tm, Tm_1^{D2} and Tm_2^{D2}. An estimate of a Tm does not neccesarily reflect a valid result. Therefore, several routines were implemented in `diffQ()` and `diffQ2()` which try to catch cases where an experiment went wrong. This includes a test if the data originate from noise, a test which analyses the difference between the approximate Tm and the calculated Tm, and a tests for the goodness of fit value ($R^2_{adj.}$, Normalized-Root-Mean-Squared-Error (NRMSE)) of the quadratic polynomial. The functions will give a warning message and point to the potential error[15]. A good practice is to visualize the output and to control the peak heights (Example in Section "Multiplex analysis of dual melting peaks"). A bimodal probe system (compare Figure 3B) requires the separation of the analysis. The minimum and maximum of the approximate first derivative have to be determined independently. Although `diffQ()` is a major function it has only a simple plot function. By setting the argument `plot = TRUE` plots for single melting curves can be investigated (Figure 9). `diffQ()` accepts further following arguments:

[12]The paramter `rsm` is avilable in both functions to double the temperature resolution. This may also reduce noise.

[13]Besides the algorithm used in `diffQ()` there are further ways to calculate the approximate derivative in R such as `diff()` (**base**) or functions from the **fda** package (Ramsay et al., 2013).

[14]Quadratic polynomials are a good compromise because they are easy to implement, do not tend to swing and fit non-linear progresses sufficiently flexible.

[15]See example for `diffQ()` in Roediger (2013).

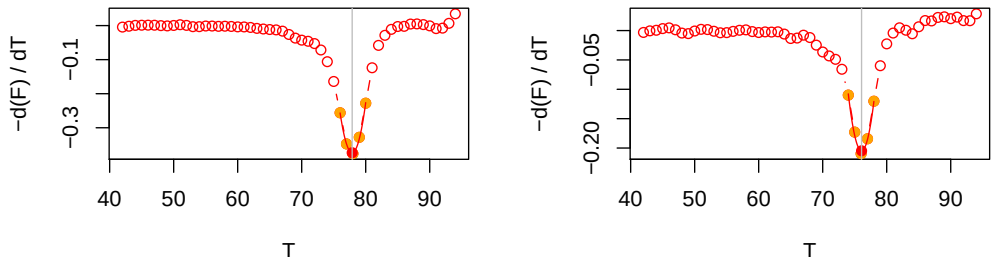

Figure 9: By default `diffQ()` shows no melting peak plot. `diffQ()`, with the argument `plot = TRUE`, can be used to show the melting peak (red dot) and fitted region from the quadratic polynomial (line) of a single melting curve. In this example *HPRT1* (left) and $MLC-2v$ (right) from `MultiMelt` are used.

- `fct` accepts `min` or `max` as argument and is used to define whether to find a local minimum ("negative peak") or local maximum ("positive peak").
- `fws` defines the number (n) of left and right neighbors to use for the calculation of the quadratic polynomial.
- `plot` defines if a single plot of the melting peak should be created. If `FALSE` (default) no plot is created.
- `negderiv` is used to change the sign of the derivatives. If `TRUE` (default) then the first negative derivative is calculated. This argument was implemented to compare different quencher / fluorophore combinations (compare Figure 2).
- Functions for a graphical output include `peak` to show the peak values and `deriv` to show the first derivative with the color assigned to `col`. `derivlimits` and `derivlimitsline` show the number of neighbors (n) or the region used to calculate the Tm. `vertiline` draws a vertical line at the Tms (Figure 10).

```
# Load MultiMelt data set.
data(MultiMelt)

# Draw figures on the graphics device in two columns.
par(mfrow = c(1, 2))

# Use mcaSmoother to check and smooth the raw data for HRPT1 (2) and MLC-2v (14)
# with the default setting.
# Plot the first derivative of the two samples.
for (i in c(2, 14)) {
  tmp <- mcaSmoother(MultiMelt[, 1], MultiMelt[, i])
  diffQ(tmp, plot = TRUE, vertiline = TRUE)
}
```

For sophisticated analysis and plots it is recommended to use `diffQ()` as part of the procedure. The arguments (e.g., `peak`, `deriv`) can be used when a plot already exists. `diffQ2()` calls instances of `diffQ()` to calculate Tm_1^{D2} and Tm_2^{D2}. The arguments are similar to `diffQ()`. Both `diffQ()` and `diffQ2()` return objects of the class `list`. Accessing components of lists is done as described elsewhere (Venables et al., 2013; Roediger, 2013) either by name or by number.

Practical applications

Multiplex analysis of single melting peaks

`diffQ2()` was used to investigate effects of the surface capture probe density which is represented by the maximal $refMFI$ value. The Tm, Tm_1^{D2} and Tm_2^{D2} values were determined simultaneously on 12 microbead populations (12-plex) either for *HPRT1* (compare Figure 8) or $MLC-2v$ using data from `MultiMelt`. In the following *HPRT1* was used as example. The corresponding matrix was called `HPRT1`[16].

```
# Load MultiMelt data set.
data(MultiMelt)
```

[16]The script for $MLC-2v$ is similar with the exception that the indices to access the elements of the `data.frame` need to be adapted accordingly.

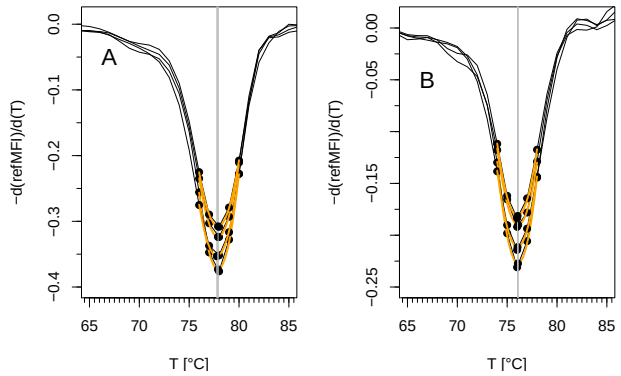

Figure 10: Output of `diffQ()` and fitted region from the quadratic polynomial (orange line) for A) *HPRT1* and B) $MLC-2v$ of four randomly selected data from `MultiMelt`.

DP	*Tm*	Tm_1^{D2}	Tm_2^{D2}
$MLC-2v$	76.08±0.04	73.99±0.11	78.51±0.13
HPRT1	77.85±0.08	75.39±0.20	80.31±0.14

Table 1: Results of Tm, Tm_1^{D2} and Tm_2^{D2} quantification for $MLC-2v$ and *HPRT1*.

```
# Create an empty matrix ("HRPT1") for the diffQ2 results (e.g., Tm).
HPRT1 <- matrix(NA, 12, 4, dimnames = list(colnames(MultiMelt[, 2:13]),
                c("Fluo", "Tm", "Tm1D2", "Tm2D2")))

# Use mcaSmoother to check and smooth the raw data. Apply diffQ2 to the smoothed data,
# calculate the values for the extreme (minimum) and assign the results to "HRPT1".
for (i in 2:13) {
  tmp <- mcaSmoother(MultiMelt[, 1], MultiMelt[, i])
  tmpTM <- diffQ2(tmp, fct = min, verbose = TRUE)
  HPRT1[i-1, 1] <- max(tmp[["y.sp"]])
  HPRT1[i-1, 2] <- as.numeric(tmpTM[["TmD1"]][["Tm"]]) # Tm
  HPRT1[i-1, 3] <- as.numeric(tmpTM[["xTm1.2.D2"]][1]) # Tm1D2
  HPRT1[i-1, 4] <- as.numeric(tmpTM[["xTm1.2.D2"]][2]) # Tm2D2
}
```

The surface capture density was determined by `max(tmp[["y.sp"]])`. Subsequently the data from the matrices `HPRT1` and `MLC2v` were plotted (Figure 11).

```
# Plot the Tm, Tm1D2 and Tm2D2 form the matrix "HRPT1" versus the surface capture
# probe density ("Fluo").
plot(HPRT1[, 1], HPRT1[, 2], xlab = "refMFI", ylab = "T [C]", main = "HPRT1",
     xlim = c(2.1, 2.55), ylim = c(72, 82), pch = 19, col = 1:12, cex = 1.8)

# Add minor tick marks to the abscissa.
require(Hmisc); minor.tick(ny = 10)
points(HPRT1[, 1], HPRT1[, 3], pch = 15)
points(HPRT1[, 1], HPRT1[, 4], pch = 15)

# Add trend lines (lm()) for the peak values.
abline(lm(HPRT1[, 2] ~ HPRT1[, 1])) # Tm
abline(lm(HPRT1[, 3] ~ HPRT1[, 1])) # Tm1D2
abline(lm(HPRT1[, 4] ~ HPRT1[, 1])) # Tm2D2
```

The melting temperature of $MLC-2v$ was 76.08±0.04 °C and 77.85±0.08 °C for *HPRT1* on all microbead populations (Table 1). This indicates that the surface capture probe density did not decrease or increase the Tm within the given range. Tm_1^{D2} and Tm_2^{D2} showed a symmetrical pattern with a low variance and therefore support that the start and end of the melting process are similar between the microbead populations. We suggest that the later peak values can be used as an additional means to describe the melting process in more detail.

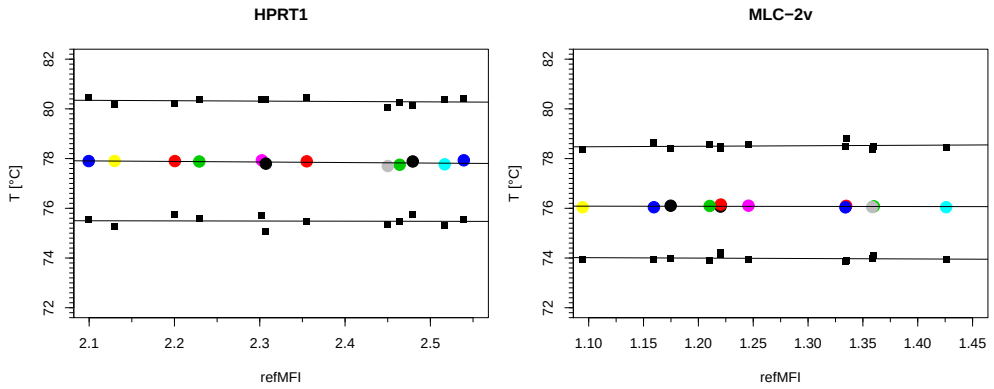

Figure 11: The peak values as function of capture probe density (sorted $refMFI$). Each colored dot represents the Tm and black squares the Tm_1^{D2} and Tm_2^{D2} of $HPRT1$ or $MLC-2v$ on one microbead population.

Multiplex analysis of dual melting peaks

The *bCP*s were hybridized with *DP*s (compare Figure 3) in order to generate bimodal melting peak patterns[17]. Due to the base composition $Poly(dA)_{20}$ *DP*s were expected to melt at lower temperatures than the *DP*s *aCS* and $MLC-2v$. First the temperature and selected probes fluorescence values from the `DMP` data set were arranged in the data frame `data.tmp` and an empty plot was created.

```
# Load DMP data set.
data(DMP)

# Use the temperature (column 1) and fluorescence (column 3, 5, 6), assign them to
# a temporary data frame and add the sample names.
data.tmp <- data.frame(DMP[, 1], DMP[, 3], DMP[, 5], DMP[, 6])
names(data.tmp) <- c("T [C]", "Poly(dA)20 & MLC-2v",
                     "Poly(dA)20 & aCS", "Poly(dA)20")

# Create a plot with the selected raw data.
plot(NA, NA, xlim = c(20, 95), ylim = c(-0.6, 0.6), xlab = "T [C]",
     ylab = "-d(refMFI) / d(T)", main = "", pch = 19, col = 1:12, cex = 1.8)

# Add minor tick marks to the abscissa.
require(Hmisc); minor.tick(nx = 10)
```

Thereafter, the data were preprocessed with `mcaSmoother()` in a loop. The arguments `bg = c(20,35)` and `bgadj` were used to adjust the background signal. This causes `mcaSmoother()` to use the subset of the data between 20 °C and 35 °C for the linear regression and background correction. To determine the Tm of the first probe (positive sign) and the second (negative sign) probe `diffQ()` was used with `min` and `max` for argument `fct`, respectively. In the loop the corresponding Tm values were assigned to the matrix `RES` and the melting curve is drawn. In addition to the lines the Tms were added (Figure 12).

```
# Create an empty matrix ("RES") for the results of the peak values (Tm) and peak
# heights (F).
RES <- matrix(NA, 3, 4, dimnames = list(colnames(data.tmp[, 2:4]),
              c("F 1", "Tm 1", "F 2", "Tm 2")))

# Use mcaSmoother to preprocess the raw data.
# Use a background correction (20-35 degree Celsius).
# Apply the smoothed data to diffQ, calculate the peak values for the extremes
# (minimum and maximum) and assign the results to the matrix "RES".
# Plot the smoothed data with the peak values and peak heights.
for (i in c(1:3)) {
  tmp <- mcaSmoother(data.tmp[, 1], data.tmp[, i + 1], bgadj = TRUE, bg = c(20, 35))
  lines(data.frame(diffQ(tmp, verbose = TRUE)["xy"]), col = i)
  RES[i, 1] <- round(diffQ(tmp, fct = max)[[2]], 2)      #fluoTm
  RES[i, 2] <- round(diffQ(tmp, fct = max)[[1]], 2)      #Tm
```

[17] See Rödiger et al. (2012b) for further details of the probe system.

DP/bCP	F_1	$Tm1$	DP/bCP	F_2	$Tm2$
$Poly(dA)_{20}$	0.59	49.6	$MLC{-}2v$	-0.57	74.7
–	0.02	91.7	$Poly(dA)_{20}$	-0.08	49.7
$Poly(dA)_{20}$	0.20	47.9	aCS	-0.15	73.9

Table 2: Tm and peak values (F_1, F_2) of the derivatives of bimodal melting curves. The detection probe and microbead bound capture probe pairs (DP/bCP) of $Poly(dA)_{20}/Poly(dT)_{20}$, $MLC{-}2v/MLC{-}2v-cap$ and $aCS/aCS-cap$ were analyzed simultaneously.

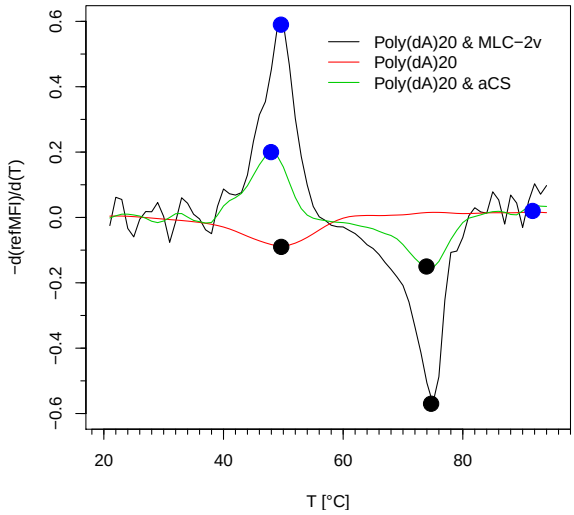

Figure 12: Melting peak pattern of $Poly(dA)_{20}$, $MLC{-}2v$ and aCS. **Note:** The sign of $Poly(dA)_{20}$ (—) is negative because the *FRET assays* was used (see Figure 2 and Rödiger et al. 2012b for details).

```
  RES[i, 3] <- round(diffQ(tmp, fct = min)[[2]], 2)      #fluoTm
  RES[i, 4] <- round(diffQ(tmp, fct = min)[[1]], 2)      #Tm
}
legend(20, 0.6, names(data.tmp[, 2:4]), lty = 1, bty = "n", col = 1:3)
points(RES[, 2], RES[, 1], pch = 19, col = 4, cex = 2)
points(RES[, 4], RES[, 3], pch = 19, col = 1, cex = 2)
```

Figure 12 shows that $Poly(dA)_{20}$ melts as single sharp peak at circa 48 °C. The detection probes aCS and $MLC{-}2v$ had a single peak at a higher Tm of circa 74 °C. Both, the number and the separation of the peaks were consistent to the estimated theoretical temperatures[18] (not shown). The results of the matrix `RES` are shown in Table 2. The Tm of $Poly(dA)_{20}$ alone and $Poly(dA)_{20}$ combined with another detection probe differed slightly. This is presumably due to different capture immobilization strategies and/or *bCP/DP* interactions (unpublished data).

Multiplex SNP detection

SNPs are important diagnostic markers for example in cardiac diseases (Villard et al., 2005; Muthumala et al., 2008). SNPs alter thermodynamic properties of dsDNA and thus the Tm. In proof-of-principle experiments melting curves were obtained from sequences of human *VIM* with a single base exchange[19]. We used the previously reported (Rödiger et al., 2012b) dual-hybridization probe on microbeads to analyze the Tm shift (inset Figure 13).

The temperature resolution was 0.5 °C per step. The `DualHyb` data were preprocessed with `mcaSmoother()` in a loop. The Tm and intensity ($fluoTm$) were stored in the matrix `RES`.

```
# Load DualHyb data set.
data(DualHyb)

# Create an empty matrix ("RES") for the results of the peak values (TmD1)
```

[18] The open source tool *PerlPrimer* by Marshall (2007) was used to estimate the theoretical temperatures.

[19] Cytosine (C) was exchanged to thymine (T) at position 41 of a 60 bp VIM oligonucleotide (see inset of Figure 13).

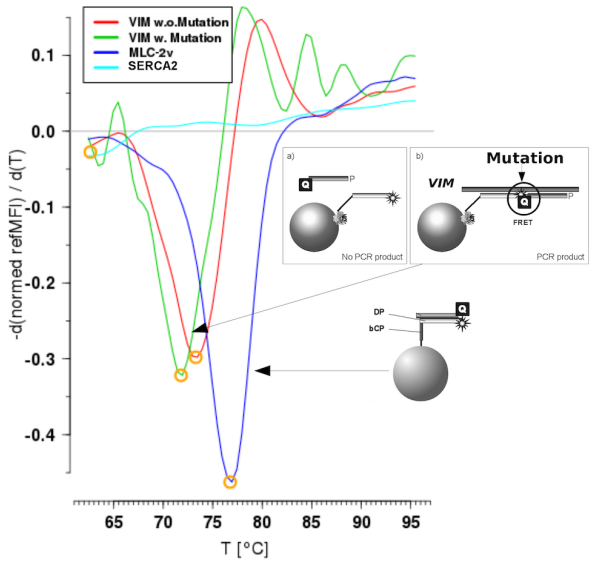

Figure 13: Proof-of-principle SNP detection with dual-hybridization assays on microbead surfaces. The melting temperature of the artificial mutated *VIM* (—) (C → T) is lower compared to the native *VIM* (—). *MLC*−2*v* (—) was used as reference for inter-assay variance. In the negative control *SERCA2* (—) no significant *Tm* was detected.

```
# and calculated peak heights (fluoTm).
RES <- matrix(NA, 4, 2, dimnames = list(colnames(DualHyb[, 2:5]),
              c("fluoTm", "TmD1")))

# Use mcaSmoother to check and smooth the raw data.
# Apply diffQ to the smoothed data, calculate the peak values for the extreme
# (minimum) and assign the results to the matrix "RES".
for (i in c(1:4)) {
  tmp <- mcaSmoother(DualHyb[, 1], DualHyb[, i + 1])
  RES[i, 1] <- round(diffQ(tmp, fct = min)[[2]], 2)     # fluoTm
  RES[i, 2] <- round(diffQ(tmp, fct = min)[[1]], 2)     # Tm
}
```

Calling `RES` gives the following output:

```
# Call RES to show the peak values and peak heights.
RES
                fluoTm  TmD1
MLC2v            -0.48 76.62
SERCA2           -0.04 62.87
VIM.w.Mutation   -0.33 71.67
VIM.wo.Mutation  -0.32 73.29
```

The algorithm calculated for the muted *VIM* a *Tm* of 71.67 °C which is 1.62 °C lower than the native *VIM* (Figure 13). This is in agreement with expected behavior. The negative control (unspecific *bCP* for *SERCA2*) had a *Tm* of 62.87 °C. But looking at the intensity showed that *SERCA2* is very low (−0.04). For screening purposes a simple cut-off would exclude this sample. The reference *MLC*−2*v* had a *Tm* of 76.62 °C. Thus the method can be applied to identify SNPs. High multiplex level was achieved by using different capture probe microbead combinations. In this example a low temperature resolution of 0.5 °C per step was used. However, since lower heating rates are positively correlated with a higher resolution for SNP analysis generally higher resolutions are recommended. The temperature resolution of the raw data can be analyzed with `diffQ(...,verbose = TRUE)[["temperature"]]`.

Summary and discussion

Experimental hardware platforms often require the development of adapted software, in particular in cases where the hardware affects the signal (e. g., photo bleaching). R is an optimal tool for such scenarios. Within this article we proposed the **MBmca** package for MCA on microbead surfaces. The

functions of the package were used for different detection-probe-systems, including direct hybridization of DNA dulexes or dual-hybridization probes for SNP detection. The capture-probe-systems produce unique melting profiles which allowed simple and rapid discrimination of different DNA sequences in one sample. The functions are useful to preprocess and inspect the data and to determine melting temperatures of immobilized DNA fragments. While used in this study for identification and quantification of biomolecules attached to microbeads it is applicable for MCA in solution too (not shown).

It was assumed that a quadratic polynomial at the approximate melting peaks can be used to calculate an accurate Tm. One drawback is that the local quadratic polynomial use a predefined number ($n = 1, \ldots, 8$) of left and right approximate melting peaks neighbors. From experience this approach proved to be reliable for the temperature resolution of $0.5 - -1$ °C per step as used in the present and the study of Rödiger et al. (2012b). Preliminary tests in solution using the LightCycler 2.0 (Roche) with high resolution (≥ 0.1 °C per step) suggest that the method works too (not shown). This implementation is designed to meet the needs of a certain experimental setup and therefore may require evaluation prior to use in new applications. Besides Tm Tm_1^{D2} and Tm_2^{D2} were proposed as additional values to describe early and late phases of the melting process on surfaces. Particularly, in investigations on the impact of the capture probe immobilization strategy or capture probe surface density these values might be useful. Methods to determine the area under the curve (AUC) were not taken into consideration due to the fact that photobleaching and quenching effects play an unknown role and are still an ongoing matter of debate in the literature. In practical terms it is recommended to implement functions from the **MBmca** package in an R GUI (see Valero-Mora and Ledesma 2012), e. g., **RKWard**[20] (Rödiger et al., 2012a). GUIs provide more intuitive means to perform multiplex high-throughput analysis with visual feedback and enrichment with additional information like goodness of fit or peak heights.

Acknowledgment

Part of this work was funded by the BMBF InnoProfile-Projekt 03 IP 611. Grateful thanks belong to the authors of the cited R packages, the R community and the **RKWard** developers.

Bibliography

M. Frasca. *delftfews: delftfews R Extensions*, 2012. URL http://R-Forge.R-project.org/projects/delftfews/. R package version 0.3-163/r164. [p42]

M. B. Gašparič, T. Tengs, J. L. L. Paz, A. Holst-Jensen, M. Pla, T. Esteve, J. Žel, and K. Gruden. Comparison of nine different real-time PCR chemistries for qualitative and quantitative applications in GMO detection. *Analytical and Bioanalytical Chemistry*, 396(6):2023–2029, 2010. URL http://www.ncbi.nlm.nih.gov/pubmed/20087729. [p37]

H. Gudnason, M. Dufva, D. Bang, and A. Wolff. Comparison of multiple DNA dyes for real-time PCR: Effects of dye concentration and sequence composition on DNA amplification and melting temperature. *Nucleic Acids Research*, 35(19):1–8, 2007. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2095797/. [p39]

C. N. Gundry, J. G. Vandersteen, G. H. Reed, R. J. Pryor, J. Chen, and C. T. Wittwer. Amplicon melting analysis with labeled primers: A closed-tube method for differentiating homozygotes and heterozygotes. *Clinical Chemistry*, 49(3):396–406, 2003. URL http://www.ncbi.nlm.nih.gov/pubmed/12600951. [p37]

F. E. Harrell Jr, C. Dupont, and et al. *Hmisc: Harrell Miscellaneous*, 2013. URL http://CRAN.R-project.org/package=Hmisc. R package version 3.12-2. [p43]

D. C. Ince, L. Hatton, and J. Graham-Cumming. The case for open computer programs. *Nature*, 482 (7386):458–488, 2012. URL http://view.ncbi.nlm.nih.gov/pubmed/22358837. [p39]

O. Marshall. Graphical design of primers with PerlPrimer. *Methods Molecular Biology*, 402:403–414, 2007. URL http://www.ncbi.nlm.nih.gov/pubmed/17951808. [p48]

A. Morin, J. Urban, P. D. Adams, I. Foster, A. Sali, D. Baker, and P. Sliz. Shining light into black boxes. *Science*, 336(6078):159–160, 2012. URL http://www.sciencemag.org/content/336/6078/159. [p39]

[20] Core structures of the **MBmca** were implemented in the **RKWard** melt plug-in (Rödiger et al., 2012a).

A. Muthumala, F. Drenos, P. M. Elliott, and S. E. Humphries. Role of beta adrenergic receptor polymorphisms in heart failure: Systematic review and meta-analysis. *European Journal of Heart Failure*, 10(1):3–13, 2008. URL http://www.ncbi.nlm.nih.gov/pubmed/18158268. [p48]

J. O. Ramsay, H. Wickham, S. Graves, and G. Hooker. *fda: Functional Data Analysis*, 2013. URL http://CRAN.R-project.org/package=fda. R package version 2.4.0. [p44]

K. M. Ririe, R. P. Rasmussen, and C. T. Wittwer. Product differentiation by analysis of DNA melting curves during the polymerase chain reaction. *Analytical Biochemistry*, 245(9):154–160, 1997. URL http://www.ncbi.nlm.nih.gov/pubmed/9056205. [p37]

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, 2008. URL http://bioinformatics.oxfordjournals.org/content/24/13/1549.abstract. [p39]

S. Rödiger, M. Ruhland, C. Schmidt, C. Schröder, K. Grossmann, A. Böhm, J. Nitschke, I. Berger, I. Schimke, and P. Schierack. Fluorescence dye adsorption assay to quantify carboxyl groups on the surface of poly(methyl methacrylate) microbeads. *Analytical Chemistry*, 83(9):3379–3385, 2011. URL www.ncbi.nlm.nih.gov/pubmed/21413805. [p39]

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, 2012a. URL http://www.jstatsoft.org/v49/i09. [p37, 50]

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. In *Molecular Diagnostics*, volume 133 of *Advances in Biochemical Bioengineering/Biotechnology*, pages 35–74. 2012b. doi: 10.1007/10_2011_132. [p37, 38, 39, 40, 47, 48, 50]

S. Roediger. *MBmca: Nucleic Acid Melting Curve Analysis on Microbead Surfaces with R*, 2013. URL http://CRAN.R-project.org/package=MBmca. R package version 0.0.2-1. [p40, 43, 44, 45]

P. Rousseeuw, C. Croux, V. Todorov, A. Ruckstuhl, M. Salibian-Barrera, T. Verbeke, M. Koller, and M. Maechler. *robustbase: Basic Robust Statistics*, 2013. URL http://CRAN.R-project.org/package=robustbase. R package version 0.9-10. [p40]

M. M. Sekar, W. Bloch, and P. M. St John. Comparative study of sequence-dependent hybridization kinetics in solution and on microspheres. *Nucleic Acids Research*, 33(1):366–375, 2005. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC546151/. [p37]

The signal Developers. *signal: Signal Processing*, 2013. URL http://R-Forge.R-project.org/projects/signal/. [p42]

P. M. Valero-Mora and R. Ledesma. Graphical user interfaces for R. *Journal of Statistical Software*, 49(1):1–8, 2012. URL http://www.jstatsoft.org/v49/i01/paper. [p50]

W. N. Venables, D. M. Smith, and the R Core Team. *An Introduction to R*. R Foundation for Statistical Computing, Vienna, Austria, 2013. URL http://www.CRAN.R-project.org/doc/manuals/R-intro.pdf. [p40, 45]

E. Villard, L. Duboscq-Bidot, P. Charron, A. Benaiche, V. Conraads, N. Sylvius, and M. Komajda. Mutation screening in dilated cardiomyopathy: Prominent role of the beta myosin heavy chain gene. *European Heart Journal*, 26(8):794–803, 2005. URL http://www.ncbi.nlm.nih.gov/pubmed/15769782. [p48]

A. Willitzki, R. Hiemann, V. Peters, U. Sack, P. Schierack, S. Rödiger, U. Anderer, K. Conrad, D. P. Bogdanos, D. Reinhold, and D. Roggenbuck. New platform technology for comprehensive serological diagnostics of autoimmune diseases. *Clinical & Developmental Immunology*, 2012:1–8, 2012. URL https://www.ncbi.nlm.nih.gov/pubmed/23316252. [p38]

A. Zeileis and G. Grothendieck. zoo: S3 infrastructure for regular and irregular time series. *Journal of Statistical Software*, 14(6):1–27, 2005. URL http://www.jstatsoft.org/v14/i06. [p42]

L. Zhou, L. Wang, R. Palais, R. Pryor, and C. T. Wittwer. High-resolution DNA melting analysis for simultaneous mutation scanning and genotyping in solution. *Clinical Chemistry*, 51(10):1770–1777, 2005. URL http://www.ncbi.nlm.nih.gov/pubmed/16189378. [p37]

Stefan Rödiger
Charité-Universitätsmedizin Berlin
and
Lausitz University of Applied Sciences
Faculty II
Germany
Stefan.Roediger@HS-Lausitz.de

Alexander Böhm
Lausitz University of Applied Sciences
Faculty I
Germany
Alexander.Boehm@HS-Lausitz.de

Ingolf Schimke
Charité-Universitätsmedizin Berlin
Germany
ingolf.schimke@charite.de