Big picture stuff

Reviewers:

Important correspondence:

1.) We don’t want to mess with published data

Brent: What melt data should we use? My main thought here is that I don't want to draw negative attention to any of our previous melt data...where people begin to question the data we and others have already published using Meltwin. I'm not sure how to handle this. Some possibilities would include (1) using unpublished data but not provide sequences (maybe just refer to them as Duplex 1, not provide raw data, etc.) so we could publish later in a thermo paper (not sure if that's legit), (2) using previously published data if the free energy values are very similar and we're just focusing on smaller error (again...not drawing too much negative attention to previously published work), or (3) collect new data that will not be part of another project (which will take more time and have some cost associated with it). I can see how a direct comparison to Meltwin could be a good thing for MeltR, but I'd hate to do that at the expense of all of the data previously published using Meltwin. This is my major concern. Is there a way to tout the data analysis of MeltR without bringing down Meltwin? Thoughts?

Phil: I think we can say only the nicest things about Meltwin. Nothing disparaging at all. At the same time, we can talk about reducing error and improving accuracy going forward. We would be very careful to say that prior data is still valuable.

Jacob: I agree with Phil. We say, “Meltwin is great. It enabled a generation of nearest neighbor parameters and all the tools that rely on them. Nobody can get Meltwin anymore and its a black box.” We can leave it at that.

On not drawing negative attention to published data, this is important. Any differences between the analysis below and the published analysis are do to idiosyncrasies. I want to proceed with the data we have. However, this is ultimately, not our data. If we are nearing a complete draft and all authors are not comfortable, I will collect a new data set to use for this paper.

2.) How do justify baseline trimming versus fitting the whole data set?

Brent: If a lot of the focus is on the baseline trimming tool, we need to be able to convince people (who think like Phil) that there are good reasons to trim.

Phil: I think we should allow different options for trimming but my philosophy is the baselines as long as possible but not longer option. This might do something recursive like look for the first five data points away from the transition that fit y=mx+b with random residuals and then lengthen 1 point at a time, stopping when the residual lose randomness.

Jacob: There are good reasons to trim. First of all, everyone does it. Second, linear baselines are an approximation and this approximation gets less valid the longer the baseline is. The question is “how to trim?” and everyone has to make their own idiosyncratic rules, which are always hard to justify because every reviewer can say “what about this?”. My feeling is that people would be happy for us to take this out of their hands if we present a sufficiently clever method.

I really liked Phil’s idea for the recursive residual analysis. Unfortunately, it does not work. (1) The baselines of ideal melting curves are never perfectly linear because it is a sigmoid. (2) On real data, this strategy just starts grabbing random parts of the melting curve.

3.) Meltwin exists and is trusted, how do we justify MeltR

Brent: I'm not sure what you had in mind for the paper. At first, I was thinking about all of the comparisons that could be made to Meltwin. But now, the more I think about it, I'm not sure that is necessary. Not many people use Meltwin, so a point-by-point comparison is probably not needed, as MeltR would be the only software available for this purpose (unless you know of others). But, I could work with Sebastian to come up with a list of limitations related to Meltwin that aren't related to the data analysis (not available for distribution, some features don't work on newer operating systems, etc.) that could be discussed in the paper.

Phil: This is more tricky. Again, I think we should cherish Meltwin and the many important papers published with it.

Jacob: I want a direct comparison because it will make Meltwin users feel safe and I want to use the word “Meltwin” a lot because it will make our paper pop up on google when somebody does a melting curve, reads a Znosko paper, and starts looking for a copy of Meltwin.

Introduction to do list

- Draft

Results to do list

- Plot the H and S instead of the H and Tm on Figure 2A-C
- Add the error quantile argument to figure 2D
- Add a plot that more succinctly summarizes percent error between methods to Figure 4
- Collect data on a non-two state folding helix for Figure 5

Methods to do list

- Theory section for the baseline trimmer in the manual
- Transfer over to the methods

Discussion to do list

- Make a list of points we want to circle back to
- outline

SI to do list

- ~~Supplemental_file_1.pdf: Supplemental figures and tables~~
- ~~Supplemental_file_2.pdf: meltR.A help documentation~~

Coding to do list

- ~~Check pH extinction coefficient options – this option does not exist~~
- ~~Add an easy outlier exclusion protocol to meltR.A?~~
- Test easy outlier exclusion protocol to meltR.A?
- ~~Iron out parallel code for multicore computers~~
- Expand description of the outputs in the help file

MeltR: A Software Package that Provides Facile Determination of RNA Folding Energies from Absorbance Data

Jacob P. Sieg[1,2], Sebastian Arteaga, Brent Znosko, Philip C. Bevilacqua[1,2,3]

[1]Department of Chemistry, Pennsylvania State University, University Park, PA 16802.

[2]Center for RNA Molecular Biology, Pennsylvania State University, University Park, PA 16802.

[3]Department of Biochemistry and Molecular Biology, Pennsylvania State University, University Park, PA 16802.

Keywords: Absorbance melting curves, folding energies, RNA thermodynamics

Abstract

UV-detected absorbance melting curves of RNA are used to determine helix folding energies, providing the thermodynamic insight into RNA secondary structure that underlies a plethora of structure prediction tools. Appropriate analysis of absorbance melting curves is not trivial, requiring many data preprocessing, regression, and error analysis steps. The absorbance melting curve-fitting software *Meltwin*, originally introduced in 1996, provided researchers with a consistent and facile absorbance melting curve analysis platform that provided a generation of RNA folding parameters. Unfortunately, *Meltwin* software, has non-transparent algorithms, is no longer maintained, and relies on idiosyncratic choices by the user. Herein, we provide *MeltR*, an open-source, curve-fitting package in the popular R statistical programming language. The *meltR.A* function of *MeltR* provides the same facile conversion of absorbance data to folding energies provided by *Meltwin*, but has additional useful features for experiments. In particular, the *BLtrimmer* function provides a consistent protocol that improves folding energy estimation and uncertainty analysis, based on analysis of an ensemble of randomly trimmed baselines. We believe that *MeltR* will be a useful tool for analyzing another generation of absorbance melting curve experiments.

Introduction

Paragraph 1: Absorbance melting curves are important

Paragraph 2: Fitting is tedious (brief summary of the established method)

Paragraph 3: Meltwin is nice

****Maybe there should be a paragraph in here justifying the need for MeltR. Like, what questions still remain that need to be answered with this analysis of melting curves?****

Paragraph 4: Present MeltR, MeltR is an R package, and its advantages

Using MeltR requires a small amount of knowledge of the R programming language. This knowledge is easily picked up from the extensive help documentation, manuals, and video tutorials provided at https://github.com/JPSieg/MeltR to help new users. A small number of R specific words that are helpful for the reader of this paper are underlined and definitions are provided at the end of the text.

Results

Facile fitting of raw absorbance data with meltR.A

MeltR includes the function meltR.A, which provides the same analysis as Meltwin and a simple usage (Figure 1A). The meltR.A function requires three user defined arguments. The first essential argument provides the data set, which is produced by reading a tidy formatted, delimited text file into a R data frame (Figure 1A.1). The second essential argument, NucAcid, specifies the nucleic acid species in the melt, provided in R as a vector, and is used to determine the extinction coefficient of the nucleic acid and subsequently the nucleic acid strand concentration (Figure 1A.2). The first element of the vector is the type of nucleic acid, either “DNA” or “RNA”, and the remaining elements are the sequences in the sample. Alternatively, the first element can be “custom”, and the remaining elements are custom extinction coefficients in $M^{-1}cm^{-1}$. The third element is the molecular model of the duplex, either "Monomolecular.2State", "Heteroduplex.2State", or "Homoduplex.2State", for a single self-structured strand, two non-self-complementary strands forming a duplex, or a single, self-complementary strand forming a duplex (Figure 1A.3). The ”meltR.A” function has two other important arguments, the “concT”, or the concentration used to calculate the strand concentration (Figure 1A.4), and “fitTs”, or the temperature range used in the fit (Figure 1A.5). The fitTs argument is used for baseline trimming (discussed in detail below), and can be a vector containing the high and low for the temperature range to be fit for all samples. Alternatively, the fitTs argument can be a list of vectors specifying temperature range for each sample.

The meltR.A function provides three analysis (Figure 1B). The first analysis, is method 1, individual fits, which fits each sample to an individual regression model to calculate thermodynamic energies (Figure 1B.1), and reports the average of thermodynamic energies over the samples (Figure 1B.2). The second analysis is method 2, which fits the Van’t Hoff relationship between the T_m and the strand concentration (C_t) from different samples to a regression model, and calculates thermodynamic energies (Figure 1B.2). Methods 1 and 2 were both provided by Meltwin. The meltR.A function also provides a third analysis, not provided by Meltwin, Method 3 Global fitting (Figure 1B.2). Global fitting combines method 1 and method 2, by directly fitting the raw absorbance melting curves, similar to method 1, but uses information from every sample, similar to method 2. Figure 1B shows the response generated when a user fits a data set with meltR.A. Note, the response includes the fractional error between methods (Figure 1B.3), which is the range of each folding energy calculated for methods 1, 2, and 3, divided by the average. This metric is useful for assessing fits.

The meltR.A function saves the result of a fit to a number of PDF figures and comma separated value (CSV) spreadsheets, to help the user assess and present results (SI Figure 1). The CSV spreadsheets can be opened with Excel or passed to other data analysis programs. The first two files are PDF depictions of the quality of the fits for method 1, where a good fit shows agreement between the data and the fits, depicted as black circles and red lines, respectively (SI Figure 1A-B). The third file is a CSV containing the results for the individual fits in Method 1 (SI Figure 1C). The fourth file is a PDF depiction of the data and fit for method 2, where a good fit exhibits a linear relationship between $1/T_m$ and $\ln C_t$ (SI Figure 1D). The fourth and fifth files are PDF depictions of the quality of the fits for method 3, where a good global fit shows agreement between the data and the fits, depicted as black circles and red lines, respectively (SI Figure 1E-F). The last file is a CSV summarizing the folding energies calculated using all three methods (SI Figure 1G).

A number of other features of meltR.A are described in the help file (SI file 2). Notably, meltR.A settings can be changed to use an alternative method to calculate the T_m for each sample, calculate extinction coefficients at different wavelengths and pHs, and to turn off methods 2 and 3. Turning off methods 2 and 3 is useful for experiments that compare absorbance melting curves between conditions. Likewise, meltR.A generates an extensive fit object in R, consisting of a list of data and statistics. The meltR.A fit object is included for advanced analysis of the data, including regression statistics, residuals, and useful transformations of the data such as first and second derivatives. This fit object includes more information than was provided by Meltwin. Elements of the meltR.A fit object can be written to text files for analysis in other programs, or passed to other R based analysis such as the BLtrimmer in MeltR (see below).

Fit parameters are sensitive to baseline trimming

Baselines are approximated as linear in melting curve analysis in both Meltwin and MeltR. The linear assumption is approximately true, due to small changes in base stacking in the duplex and single stranded state with temperature. This approximation has no physical basis and is quantitatively accurate over short temperature ranges, meaning that baselines should be trimmed to include no more data than is required to define the duplex and single stranded state. Aberrant data at low temperatures due to condensation on the cuvette or aberrant data at high temperatures due to sample evaporation should be trimmed. Baselines should also be trimmed to compensate for small internal aberrations due to instrumental errors or air bubbles. Accordingly, Meltwin and MeltR provide the capability to trim baselines by specifying a temperature range to fit. However, users must make idiosyncratic choices about how to trim manually, as no protocols have been established. We thus sought to provide an auto-baseline trimmer in MeltR, that gives users a consistent, rational, and statistically appropriate baseline-trimming protocol.

To first understand how the length of baselines affects the accuracy of folding energy determination, we first fit modeled data (Figure 2A top panel, black circles). Data was modeled assuming a ΔH° of -64.76 kcal/mol, a Tm of 46.4 °C, and a Ct of 8 μM, a random absorbance error of 0.001, and realistic baseline slopes and intercepts. The range where the helix was between 90% folded and 10% folded was identified to be the upper and lower bounds of the low temperature and high temperature baselines respectively. Then, base lines were added in 1 °C increments, and the data was fit to determine the relationship between fit accuracy and baseline length (Figure 2A top panel, colored lines). The resulting ΔH° and Tm produced by these fits were inaccurate for baselines below 5 °C in length, but converged on the correct value at higher temperatures (Figure 2A, middle and bottom panels). Thus, analysis of modeled data indicated that baselines should be no shorter than 5 °C.

We next analyzed the effects of baseline length on folding energy determination on real data. We started on a nearly ideal, real melting curve (Figure 2B top panel, black circles). We first estimated the temperature range where the helix was between 90% and 10% folded, using a fit of the un-trimmed curve. Then we added baselines in 1 °C increments and fit the data. The resulting energies produced by these fits were inaccurate below 5 °C, but converged on the correct value at higher temperatures (Figure 2B, middle and bottom panels). Next, we analyzed the he effects of baseline length on folding energy determination on a non-ideal, real melting curve. Once again, the ΔH° and T_m produced was inaccurate for baselines that were less than 5 °C. However, the ΔH° did not converge on a single consistent value (Figure 2B, middle panel).

The data is real, so the true ΔH° cannot be known. However, a user could identify an optimum baseline range by iteratively fitting the data using different baseline ranges with Meltwin or meltR.A. The results for this curve could be compared with the results for the other curves in the data set, then the user could choose the baseline range that provides the best internal consistency across the data set. This approach would lead to a more accurate estimate of the folding energy from a data set, but is time consuming, idiosyncratic, and incorporates a systematic error (the choice of baseline range to use in the analysis), that is difficult to treat statistically.

Thus, MeltR provides a baseline trimmer function, the “BLtrimmer”, that works on three principles:

1. A large number of baseline ranges should be randomly generated and subsequently fit. These baseline ranges should be between 5 and 25 °C, lower and higher than the temperatures where the sample is 90% and 10% helical, for low temperature and high temperature baselines respectively. The 5 °C lower limit on baseline length ensures that the baseline ranges can produce accurate folding energies (Figure 2A). The 25 °C upper limit on baseline length ensures that the baseline range minimizes violation of the linear approximation.
2. The best combinations of baseline ranges produce the most internally consistent folding energy estimates across the data set, as assessed by agreement between ΔH° produced by different samples in method 1, and by agreement between the average ΔH° produced by method 1 and 2.
3. Besides principle 1 and 2, the 25% of baseline ranges that produce the most internally consistent folding energies should be treated as an ensemble of equally feasible folding energies. This converts baseline trimming, a systemic error that cannot be treated statistically, into a random error which can be treated statistically.

The BLtrimmer treats fit results from an optimal ensemble of baseline combinations statistically

This section introduces the BLtrimmer function in MeltR, demonstrates it’s usage and results (Figure 2D-E), and describes its internal function. The next section examine the accuracy of the BLtrimmer fitting modeled, published, and unpublished data.

The BLtrimmer has only one required argument, a MeltR fit object produced by meltR.A (Figure 2D.1). This MeltR fit object is used to estimate the temperatures where the sample is 90% and 10% helical (Figure 3A) in order to identify the core of the melting curve, which is not trimmed (Figure 3B). The other arguments shown in Figure 2D are adjustable but set to the recommended value by default. The first optional argument is the number of ranges to generate per sample (Figure 2D.2). The second optionally adjustable argument is the temperature step between baselines (Figure 2D.3). Thus, by default, the BLtrimmer will produce 5 symmetrical baselines per sample, at lengths of 5, 10, 15, 20, and 25 °C (Figure 3C). The third optionally adjustable argument is the number of baseline combinations to test (Figure 2D.4). The BLtrimmer exhaustively permutes each baseline range with every other baseline range from every other sample. By default, this will produce 5^N combinations, where N is the number of samples in the data set. Testing all combinations would require a lot of computational power and is not necessary. One thousand randomly selected baseline combinations are enough to explore the error space, and take 1-3 minutes to test on a laptop computer (Figure 3D).

The last important argument is the error distance threshold, which by default selects the 25% of baseline combinations that produce the most internally consistent folding energies (Figure 2D.4). The error distance is a statistic that equally combines agreement between the ΔH° produced by different samples by method 1, and agreement between the ΔH° produced by method 1 and 2. Data from each baseline combination are first fit with method 1 and 2, and the normalized standard deviation of method 1 ΔH° values and the normalized difference between method 1 and 2 ΔH° values are calculated (Figure 3E). These values are then ranked and quantilized, placing each value on a relative scale from 0 to 1, where 0 is the most accurate and 1 is the least accurate (Figure 3F). The error distance is then calculated as the Pythagorean distance between a point on the plot and the origin, thus, equally accounting for agreement for method 1, and by agreement of ΔH° produced by method 1 and 2. Then the BLtrimmer selects the 25% of baseline combinations that produce the smallest error distance (Figure 3G). The error distance threshold can be decreased to be more selective, or increased to 100% to generate an exhaustive description of the dependence effects of folding energies on baseline trimming. Lastly, the BLtrimmer passes the ensemble of internally consistent baseline combinations back to meltR.A for fitting. The results are a averaged and 95% confidence intervals are determined (Figure 3H).

The BLtrimmer function saves the results as a CSV and a PDF. The CSV consists of a table of folding energies and confidence intervals, determined from the ensemble of baseline combinations using the three methods (Figure 2E, SI table 1). The PDF contains plots that the user can use to assess the results of a BLtrimmer run (SI Figure 2). The first two plots are histograms showing the distribution of normalized standard deviation of method 1 ΔH° values (SI Figure 2A) and the normalized difference between method 1 and 2 ΔH° for all of the baselines tested (SI Figure 2B) . Blue lines represent the 25% of baseline combinations that produce the most internally consistent folding energies. The peak of these distributions should occur below 15% for two state folding sequences (See below). The next plots shows the quantilized standard deviation of ΔH° values in the method 1 fit versus the quantilized difference between ΔH° values in the method 1 and method 2 fit, for each baseline combination (SI Figure 2C). Ideally, the data points should be evenly spread throughout the space of the graph, as clustering indicates fitting artifacts due to aberrant data. The raw values are plotted below the quantilized graph for the convenience of the user (SI Figure 2F). The fourth plot shows the average ΔH° versus the standard deviation of ΔH° calculated from method 1, and should exhibit no trend or clustering (SI Figure 2D). Lastly, the fourth plot shows the average enthalpy produced by method 1 (Black) or the enthalpy produced by method 2 (Red), versus the difference in ΔH° between methods 1 and 2, for each baseline combination. For two state folding sequences, these data should exhibit no trend and the black and red clusters should overlap (See below).

MeltR accurately reproduces thermodynamic parameters

We first determined that MeltR accurately determined folding energies by fitting modeled data. Nine realistic absorbance melting curves, representing a thousand-fold concentration range, were modeled for 90 two-state folding helices. Slopes, baselines, and absorbance errors were randomly generated. The modeled data were then fit with meltR.A followed by the BLtrimmer, and using no manual baseline trimming. BLtrimmer results for all three methods were in good agreement with the known folding energies (SI figure 3). Errors were evenly distributed between being higher than the known value and lower than the known value, indicating that the BLtrimmer was not incorporating a systematic error. Interestingly, method 2 was less accurate than methods 1 and 3, with and average ΔH° error of 3.3%, 0.4%, and 0.6% respectively (see discussion).

We next tested MeltR with little user specification of baseline ranges by fitting real data. We compiled 5 data sets collected on self-complementary RNA from a published source,I and 6 new data sets collected on a non-self-complementary RNA duplex are presented here (SI Table 2). The data were fit with meltR.A, only using a fixed baseline trim to remove aberrant data from the extremes of the data set, when appropriate. These fits were passed to BLtrimmer, resulting in folding energy estimates using the three aforementioned methods. The three methods were in good agreement, with an average ΔH°, ΔS°, ΔG°$_{37}$ and T$_{m}$ error between methods of 4.8% (max 15.7%), 5.4% (max 17.1%), 1.2% (max 2.6%), and 0.4% (max 0.7%), respectively.

To obtain an independent measure of accuracy, we compared the folding energy we calculated with MeltR to folding energies calculated with Meltwin. Folding energies calculated with methods 1 and 2 using Meltwin were obtained from the published source for the self-complementary helices. Folding energies calculated with methods 1 and 2 were obtained using Meltwin by SA without input from JPS. MeltR method 1, with little manual trimming of baselines, was in good agreement with Meltwin method 1, with and average ΔH°, ΔS°, ΔG°$_{37}$, and T$_{m}$ error between programs of 1.8% (max 4.7%), 2.0% (max 5.7%), 2.5% (max 4.4%), and 2.2% (max 3.8%), respectively (SI Table 3). Likewise, MeltR method 2 was in good agreement with Meltwin method 2, with and average ΔH°, ΔS°, ΔG°$_{37}$ and T$_{m}$ error between programs of 3.0% (max 7.0%), 3.3% (max 7.7%), 2.6% (max 3.6%), and 2.2% (max 3.8%), respectively (SI Table 4). These errors are on average smaller than the error expected for data produced in different labs, due systematic errors in instrument calibration and data analysis (see discussion).

Interestingly, MeltR method 3 reasonably reconstituted both Meltwin method 1 and 2 (Figure 4). MeltR method 3 was in good agreement with Meltwin method 1, with an average ΔH°, ΔS°, ΔG°$_{37}$ and T$_m$ error between programs of 1.5% (max 4.4%), 1.6% (max 5.4%), 2.6% (max 5.6%), and 2.3% (max 3.6%), respectively (SI Table 5). Likewise, method 3 was in similar agreement with MeltR method 2, with an average ΔH°, ΔS°, ΔG°$_{37}$ and T$_m$ error between programs of 1.9% (max 6.3%), 2.2% (max 7.4%), 1.4% (max 3.0%), and 1.2% (max 1.8%), respectively (SI Table 5). In summary MeltR method 3 reproduces Meltwin methods 1 and 2, as well or better than MeltR methods 1 and 2. Melthod 3 in MeltR may be able to reproduce Meltwin methods 1 and 2 because it combines the raw data fitting provided by method 1 with the linking of samples using the Van't Hoff equation provided by method 2.

The BLtrimmer provides a two-state folding test

Helices are tested for two-state folding by comparing the ΔH° differences between methods 1 and 2. If the difference is smaller, or larger, than 15% of the mean value, the helix is considered two-state, or non-two-state, respectively. However, the results of methods 1 and 2 are both sensitive to baseline-trimming. This leads to a problem, its hard to tell if the helix is truly two-state folding, or if errors in baseline trimming let to the appearance of two-state folding. The baseline trimmer provides a solution to this problem, based on analysis of the distribution of helix energies derived from randomly trimmed baselines.

Paragraph 4: Histogram analysis followed by quartile analysis.

Paragraph 4: Why the BLtrimmer does not force a two state interpretation on non-two state RNA.

Discussion

Materials and Methods

Data transparency and reproducibility

Absorbance melting curves

Paragraph 1: How Brent and Sebastian compiled Adams data

Paragraph 2: Acquisition in the Bevilacqua lab

Fitting absorbance melting curves

This will be copied and cleaned up from the theory section of the MeltR manual

BLtrimmer

Modeling absorbance melting curves

This will be copied and cleaned up from the theory section of the MeltR manual Figure 4

Definitions

Function: A block of code that automates a common task in a general way. Argument: The an input a function needs to run. Tidy formatted: A consistent way to organize data a data table. Each variable must have its own column. Each observation must have its own row. Each value must have its own cell. Data frame: Practically, a table of data in R. Specifically, a list of vectors, each containing a different variable. All vectors must be the same length in a data frame. Vector: Basic unit of data storage in R. Created with the "c" function. For example, to create a vector for the NucAcid argument for meltR.A, one can use the "c" function, "c("RNA", "AAAA", "UUUU")". List: A more advanced type of data storage in R. Can be a list of anything, including vectors, data frames, and other lists.

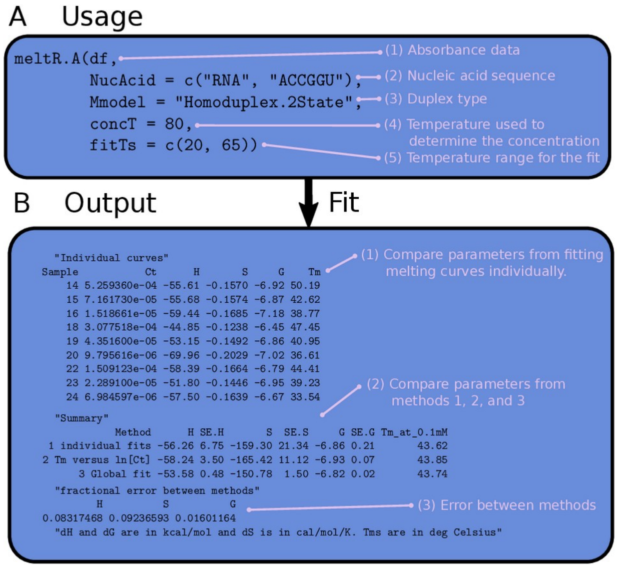
Acknowledgments

References

Tables

Table 1 Regression equations used to determine folding energies in meltR.A. (to be associated with the methods).

Figures and legends



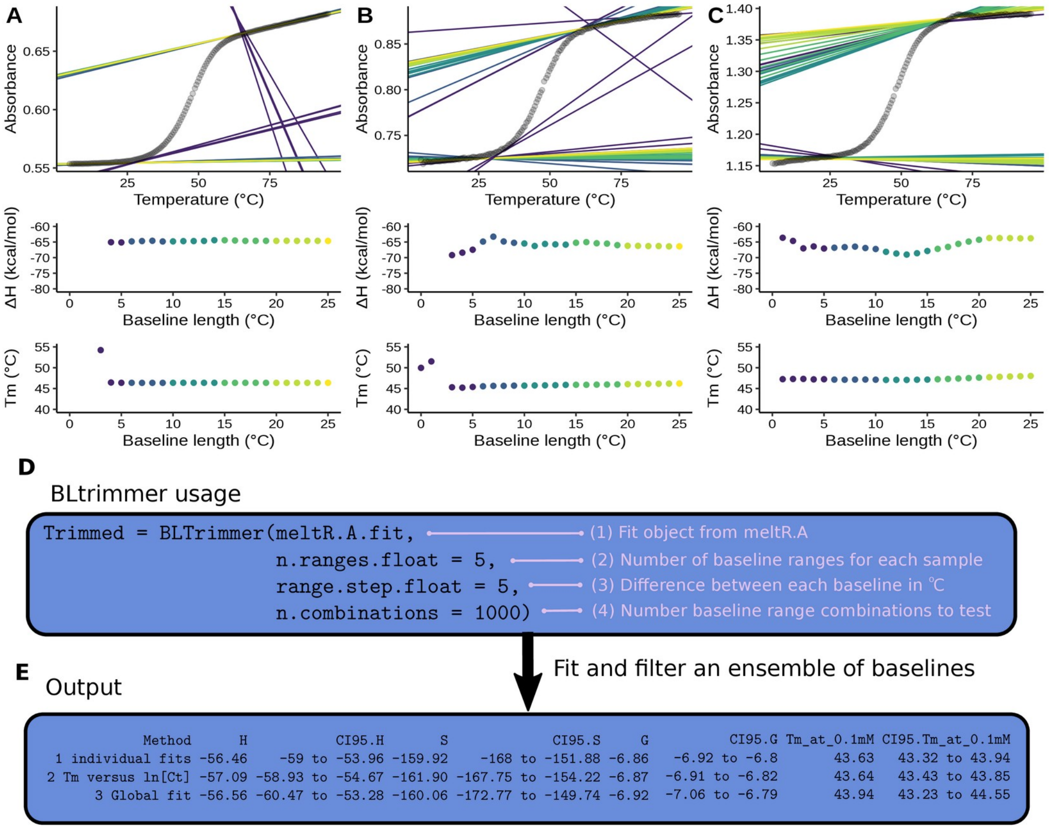
```
meltR.A(df,
        NucAcid = c("RNA", "ACCGGU"),
        Mmodel = "Homoduplex.2State",
        concT = 80,
        fitTs = c(20, 65))
```

(1) Absorbance data
(2) Nucleic acid sequence
(3) Duplex type
(4) Temperature used to determine the concentration
(5) Temperature range for the fit

```
  "Individual curves"
Sample          Ct      H       S     G    Tm
    14 5.259360e-04 -55.61 -0.1570 -6.92 50.19
    15 7.161730e-05 -55.68 -0.1574 -6.87 42.62
    16 1.518661e-05 -59.44 -0.1685 -7.18 38.77
    18 3.077518e-04 -44.85 -0.1238 -6.45 47.45
    19 4.351600e-05 -53.15 -0.1492 -6.86 40.95
    20 9.795616e-06 -69.96 -0.2029 -7.02 36.61
    22 1.509123e-04 -58.39 -0.1664 -6.79 44.41
    23 2.289100e-05 -51.80 -0.1446 -6.95 39.23
    24 6.984597e-06 -57.50 -0.1639 -6.67 33.54
  "Summary"
            Method      H SE.H       S  SE.S     G SE.G Tm_at_0.1mM
 1 individual fits -56.26 6.75 -159.30 21.34 -6.86 0.21       43.62
2 Tm versus ln[Ct] -58.24 3.50 -165.42 11.12 -6.93 0.07       43.85
      3 Global fit -53.58 0.48 -150.78  1.50 -6.82 0.02       43.74
  "fractional error between methods"
         H          S          G
0.08317468 0.09236593 0.01601164
  "dH and dG are in kcal/mol and dS is in cal/mol/K. Tms are in deg Celsius"
```

(1) Compare parameters from fitting melting curves individually.
(2) Compare parameters from methods 1, 2, and 3
(3) Error between methods

Figure 1 The usage and output of meltR.A in the MeltR package. **(A)** The usage of meltR.A in an R script. (1) The data frame (df) containing the absorbance data, created by reading a flat text file. (2) A vector containing the nucleic acid type and sequence, used to calculate extinction coefficients. Alternatively, custom extinction coefficients can be provided (Supplemental file 2). (3) Duplex type, either "Monomolecular.2State", "Heteroduplex.2State", or "Homoduplex.2State". (4) The temperature used to calculate the RNA concentration using Beer's law. (3) The temperature range that is fit. Used for manual baseline trimming. Either a vector specifying a single range for the whole data set, or a list of vectors, one for each sample. **(B)** The output of meltR.A. (1) Thermodynamic parameters for method 1, generated by fitting each curve in a data set individually. (2) Comparison of the results for the different methods. Method 1 averaged the results from fitting each melting curve individually. Method 2 fit the Van't Hoff relationship between the melting temperature and the RNA concentration. Method 3 combined Method 1 & 2, by using the Van't Hoff equation to globally fit the entire data set to one regression model. (3) Fractional error between methods used for a test of two-state folding. Non two-state folding generates a fractional error between methods that is greater than 0.15.



```
Trimmed = BLTrimmer(meltR.A.fit,
                    n.ranges.float = 5,
                    range.step.float = 5,
                    n.combinations = 1000)
```

```
             Method      H           CI95.H       S               CI95.S     G          CI95.G Tm_at_0.1mM CI95.Tm_at_0.1mM
 1 individual fits -56.46    -59 to -53.96 -159.92    -168 to -151.88 -6.86  -6.92 to -6.8       43.63   43.32 to 43.94
2 Tm versus ln[Ct] -57.09 -58.93 to -54.67 -161.90 -167.75 to -154.22 -6.87  -6.91 to -6.82      43.64   43.43 to 43.85
      3 Global fit -56.56 -60.47 to -53.28 -160.06 -172.77 to -149.74 -6.92  -7.06 to -6.79      43.94   43.23 to 44.55
```

Figure 2 Auto-baseline trimming absorbance data using the BLtrimmer in MeltR. **(A)** Dependence of fit parameters on baseline length for modeled absorbance data assuming a ΔH° of -64.76 kcal/mol, a T_m of 46.4 °C, and a C_t of 8 µM. Lower and upper base lines start where 90% and 10% (molar basis), respectively, of the RNA strands are in the helical state. (Top panel) Linear baseline fits superimposed on absorbance data. Colors correspond to baseline length specified in middle and bottom panels. (Middle panel) Dependence of ΔH° on baseline length. (Bottom panel) Dependence of T_m on baseline length. **(B)** Dependence of fit parameters on baseline length for a nearly-ideal, real absorbance melting curve collected on 5’CGAAAGGU3’/5’ACCUUUCG3’ at a C_t of 8 µM. **(C)** Dependence of fit parameters on baseline length for a non-ideal, real absorbance melting curve, collected on 5’CGAAAGGU3’/5’ACCUUUCG3’ at a C_t of 12 µM. **(D)** BLtrimmer usage in a R script. (1) A MeltR fit object produced by fitting raw data with meltR.A. (2) The number of baseline ranges the BLtrimmer will produce for each sample. (3) The temperature difference for each baseline produced on an

absorbance melting curve. (4) The number of baseline combinations to test in a given BLtrimmer run. **(E)** Output of the BLtrimmer providing folding energies and 95% confidence intervals.

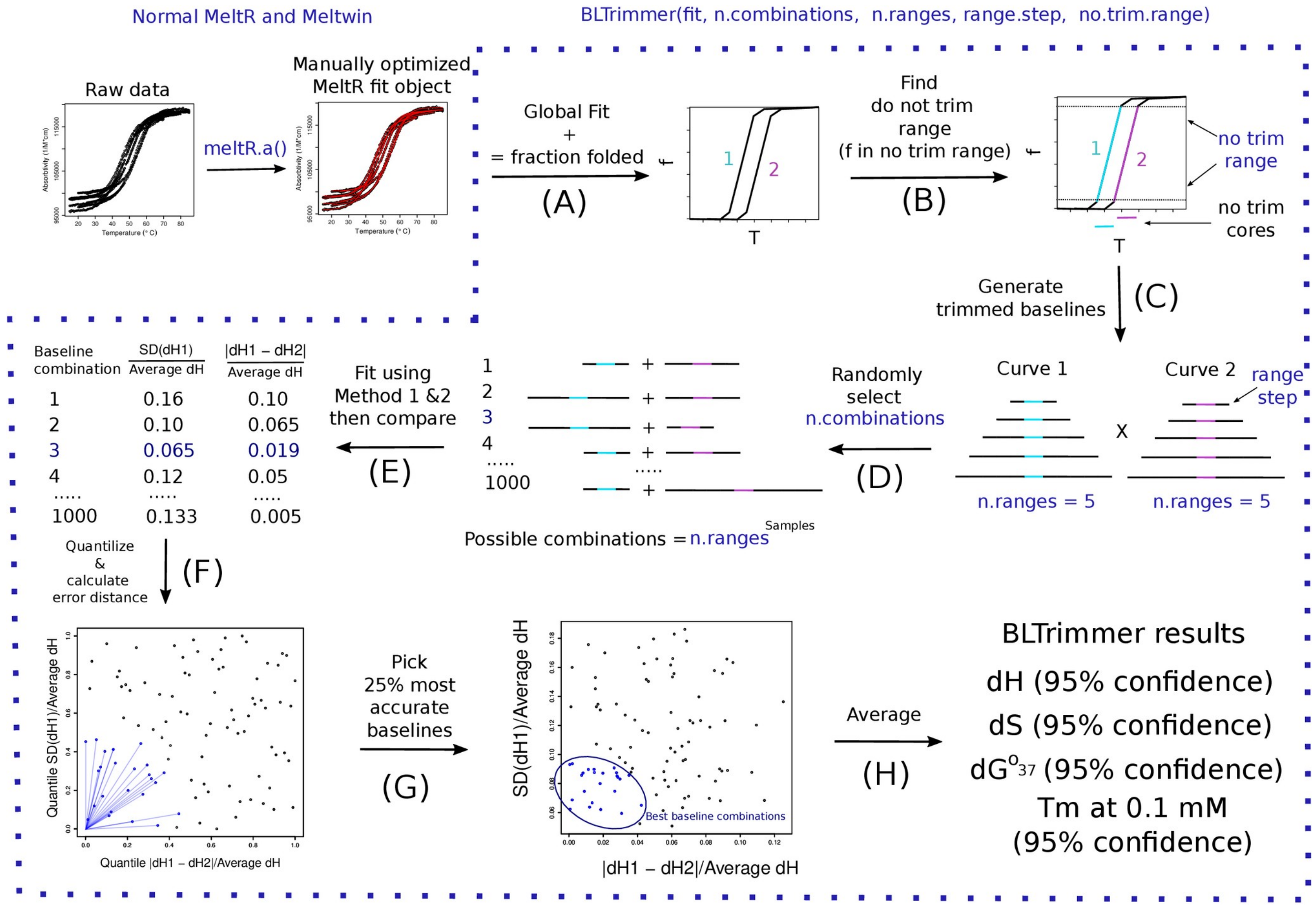


Figure 3 The MeltR BLtrimmer applies an ensemble analysis to calculate optimum helix association energies. **(A)** A MeltR fit object, produced by fitting absorbance data with meltR.A, is used to calculate the fraction of strands in the helical state (*f*) as a function of temperature. **(B)** The “no trim” temperature range is identified where *f* is greater than 0.1 and less than 0.9. **(C)** Baselines are added to the no trim range in 5 °C steps. **(D)** Ranges from each sample are exhaustively permuted to generate baseline combinations. 1000 baseline combinations are randomly selected. **(E)** Folding energies are determined for each baseline combination using method 1 (average of fitting samples individually) and method 2 (Van’t Hoff analysis). **(F)** Baseline combinations are assessed for ΔH° agreement for method 1 and ΔH° agreement between method 1 and method 2 to identify the baseline combinations that have the highest internal consistency. **(G)** The top 25% most internally consistent

baseline combinations are selected for ensemble analysis. **(H)** The ensemble of baseline combinations is passed to meltR.A and fit. The results are treated statistically to determine folding energies and 95% confidence intervals.

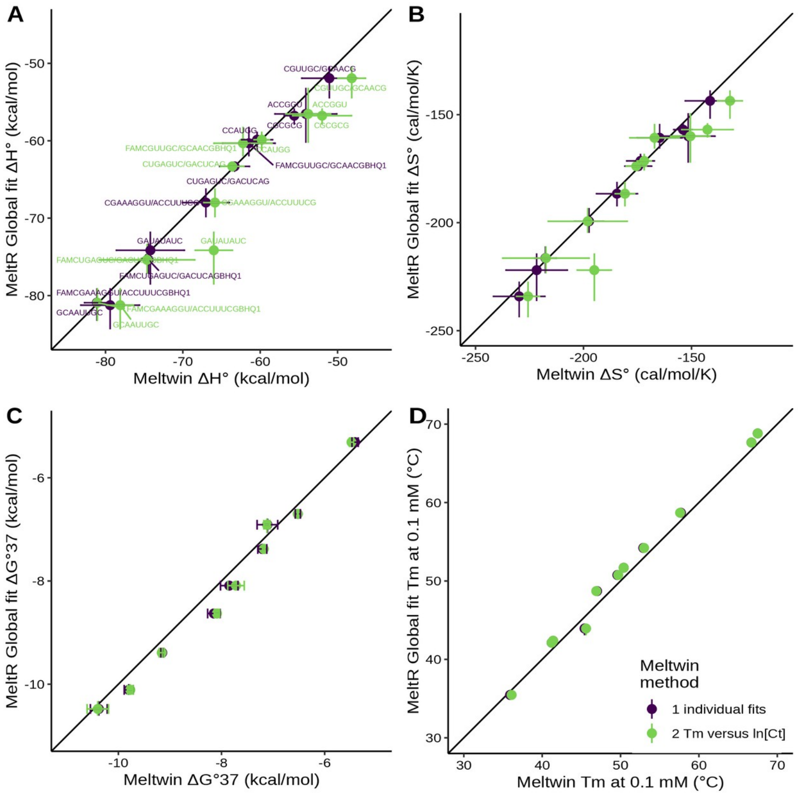


Figure 4 MeltR functions meltR.A and BLtrimmer reproduce folding energies calculated using Meltwin. **(A)** ΔH° determined from the method 3, global fitting, using MeltR versus the ΔH° determined from method 1 (purple) and method 2 (green) using Meltwin to fit the same absorbance data set. Horizontal error bars represent precision in the parameters calculated by Meltwin and vertical error bars represent 95% confidence intervals calculated using the BLtrimmer. **(B)** ΔS° determined from the method 3, global fitting, using meltR.A followed by the BLtrimmer, versus the ΔS° determined with Meltwin. Colors and error bars are the same as A. **(C)** ΔG°$_{37}$ determined from the method 3, global fitting, using meltR.A followed by the BLtrimmer, versus the ΔG°$_{37}$ determined with Meltwin. Colors and error bars are the same as A. **(D)** T_m at 0.1 mM determined from the method 3, global fitting, using

meltR.A followed by the BLtrimmer, versus the T_m at 0.1 mM determined with Meltwin. Colors and error bars are the same as A.

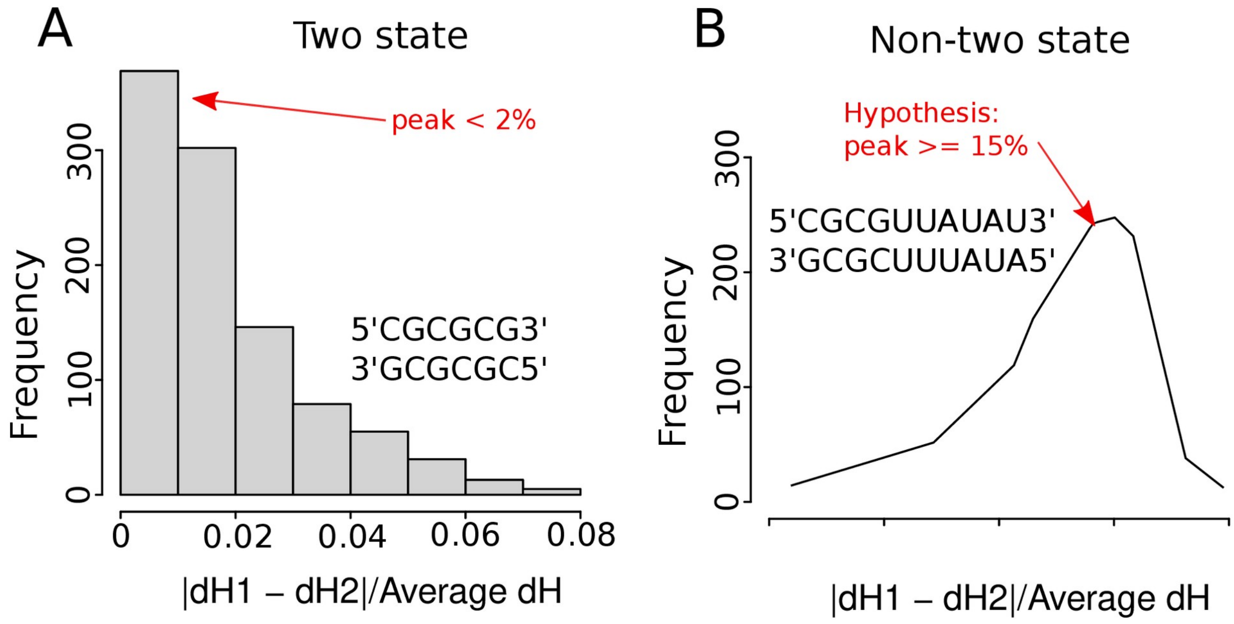


Figure 5 Two-state folding test using the BLtrimmer in MeltR. **(A)** The distribution of the error between the ΔH° using Method 1 (average of fitting samples individually) and Method 2 (Van't Hoff analysis) using 1000 randomly determined baseline combinations peaked at 2% for two-state folding sequences. **(B)** The distribution of the error between the ΔH° using Method 1 (average of fitting samples individually) and Method 2 (Van't Hoff analysis) using 1000 randomly determined baseline combinations peaked at >15% for non-two-state folding sequences.