MeltR Software Provides Facile Determination of Nucleic Acid Thermodynamics

Jacob P. Sieg[1,2,*], Sebastian J. Arteaga[3], Brent M. Znosko[3], Philip C. Bevilacqua[1,2,4,*]

[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 Chemistry, Saint Louis University, Saint Louis, MO 63103.

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

*Correspondence should be directed to jus841@psu.edu and pcb5@psu.edu

Keywords: Absorbance-detected melting curves, thermodynamics, RNA

ORCID

Jacob P. Sieg: 0000-0001-5414-1667

Sebastian J. Arteaga: 0000-0001-7641-7919

Brent M. Znosko: 0000-0002-6823-6218

Philip C. Bevilacqua: 0000-0001-8074-3434

Abstract

Thermodenaturation (melting) curves of macromolecules are used to determine folding thermodynamic parameters. Notably, this insight into RNA and DNA stability underlies nearest neighbor theory and diverse structure prediction tools. Analysis of UV-detected absorbance melting curves is complex and multivariate, requiring many data preprocessing, regression, and error analysis steps. The absorbance melting curve-fitting software *MeltWin*, introduced in 1996, provided a consistent and facile melting curve analysis platform used in a generation of folding parameters. Unfortunately, *MeltWin* software is not maintained and relies on idiosyncratic choices of baselines by the user. Herein, we provide *MeltR*, an open-source, curve-fitting package for analysis of macromolecular thermodynamic data. The *MeltR* package provides the facile conversion of melting curve data to parameters provided by *MeltWin* while offering additional features including global fitting of data, auto-baseline generation, and two-state melting analysis. *MeltR* should be a useful tool for analyzing the next generation of DNA, RNA, and non-nucleic acid macromolecular melting data.

Why it matters

Melting curves provide the input data for the thermodynamic parameters used in algorithms that predict nucleic acid structure and behavior. Analysis of melting curves is complex, requires numerous arbitrary choices by the user, and is cumbersome to perform. Herein, we introduce *MeltR*, a replacement, and improvement on the melting-curve fitting software, *MeltWin*. *MeltR* uses a facile R interface, automates data analysis, and provides analysis that is simple to distribute and reproduce. Lastly, *MeltR* can be improved and redistributed by the community without restriction.

Introduction

Thermodenaturation (melting) curves monitor macromolecular structure as a function of temperature and provide thermodynamic insight. UV-detected absorbance melting curves of nucleic acids have

facilitated extensive thermodynamic characterization of the sequence dependence of diverse nucleic acid secondary structures.[1,2] This characterization was used to develop the ubiquitous nearest neighbor model for calculating the folding energy of nucleic acid secondary structure,[3–8] which underlies widely used algorithms that predict the behavior of nucleic acids (Figure 1A).[9–15]

Extraction of parameters from melts requires many preprocessing, regression, and error analysis steps, which are cumbersome to achieve manually (Figure 1B).[16] The program *MeltWin*,[17] introduced over 25 years ago, provided researchers with a facile data analysis platform, which underlies the aforementioned characterization of nucleic acid stability. While current understanding is extensive in its scope, covering vast amounts of RNA motifs in 1 M NaCl,[8] many questions persist. In particular, thermodynamic contribution of the cellular environment[18,19] and of naturally-modified nucleotides[20] are areas of active research. Unfortunately, *Meltwin* is no longer maintained, has non-transparent algorithms, and relies on idiosyncratic choices by the user, which can bias results.

We introduce *MeltR*: an open-source package in the popular R programming language[21] that provides facile conversion of raw data to parameters. *MeltR* offers fitting protocols for fluorescence binding isotherms, as we recently reported.[22] Herein, we apply *MeltR* to fitting UV-detected absorbance melting curves of RNA, similar to *MeltWin*, describe unique automation features not provided by *MeltWin*, and benchmark the accuracy of *MeltR*. Using *MeltR* requires basic knowledge of R, which is readily acquired from the help, including video screencasts, provided at https://github.com/JPSieg/MeltR. To aid the reader, R-specific words are underlined the first time they are used, and definitions are provided.

Results

Facile fitting of raw absorbance data with meltR.A

MeltR includes the function *meltR.A*, which provides analysis comparable to *MeltWin*, and follows the flowchart presented in Figure 1B (see File S1 methods). The *meltR.A* function requires three user-defined arguments. The first provides the dataset, which is produced by reading a tidy formatted text file into an R data frame (Figure 1C.1). The second argument, “NucAcid,” specifies the nucleic acid in the melt. This is provided as a vector and is used to determine the extinction coefficient of the nucleic acid (Figure 1C.2). The first element of the vector is nucleic acid type, either “DNA” or “RNA,” and the remaining elements are the residues in the sample, which are used to calculate the extinction coefficient. Alternatively, the user can supply custom extinction coefficients or, for non-absorbance data, a series of sample concentrations. The third element is the molecular model, “Mmodel,” which can be specified as either "Monomolecular.2State,” "Heteroduplex.2State,” or "Homoduplex.2State,” for a self-structured strand, two non-self-complementary strands, or two self-complementary strands, respectively (Figure 1C.3). The *meltR.A* function has other arguments, with default settings that can be adjusted. Notably, “concT” is the temperature used to calculate the concentration (Figure 1C.4), and “fitTs” is the temperature range used in the fit (Figure 1C.5). The “fitTs” argument is used for baseline trimming and can be a vector containing the temperature range to be fit for all melts or a list of vectors specifying a temperature range for each melt.

The *meltR.A* function provides three analyses simultaneously: individual fits, $1/T_m$ versus total strand concentration (C_t), and global fitting (Figure 1D). The first analysis is individual fits (method 1), which fits each sample to an individual model (Figure 1D.1). For method 1, the averages of thermodynamic parameters from fitting each melting curve individually are reported as “Individual fits” in Figure 1D.2. The second analysis, $1/T_m$ versus the natural logarithm of C_t (method 2), fits the relationship between $1/T_m$ and ln C_t from different samples to calculate thermodynamic parameters, which are also provided in Figure 1D.2. Methods 1 and 2 were both provided by *MeltWin*.[17] The *meltR.A*

function also provides a third method, global-fitting analysis, not provided by *MeltWin* (Figure 1B). Global fitting combines features of both methods 1 and 2 by fitting the raw melting curves, like method 1, but using information from every sample in a single model, like method 2.

Figure 1D shows the response generated by *meltR.A*. Standard error (SE) values are provided for assessing the quality of the fit.[7] Note, the response also includes the maximal %error across the three methods (Figure 1D.3, File S1 methods). This metric is useful for assessing the overall quality of fits and two-state folding, where a percent error between methods >10-15% is diagnostic of multistate-folding.[2,7,16,23] Lastly, *meltR.A* can automatically save the result of a fit to a series of figures and tables (Figure S1).

Several other *meltR.A* features are described in the help file (File S2). Notably, *meltR.A* can calculate T_ms by alternate methods and calculate RNA extinction coefficients at wavelengths besides 260 nm, which can be useful as hyperchromicity is greater at 280 nm for GC-rich helices.[1] Likewise, *meltR.A* generates an extensive fit object in R, consisting of a list of data and statistics (File S2). The fit object allows advanced data analysis, including regression statistics, residuals, and data transformations such as first derivatives.

MeltR provides an automated baseline-trimming protocol

Both *MeltWin* and *MeltR* approximate the temperature dependence of the folded and unfolded states as lines.[16] These lines represent absorbance changes due to a variety of phenomena, including changes in base stacking with temperature. In general, baselines should be trimmed to exclude aberrant data and to include no more data than required to define the duplex and single-stranded states, avoiding extrapolation of baselines from temperatures far removed from the melt region. However, users generally make manual choices about trimming baselines, as no protocols have been established.[24]

One approach for baseline trimming is including a constant window flanking the melt region, for example, 20 °C on either side. In a second approach, a user could customize each melt by using different

trims. The user often chooses the trim for each melt that provides the best agreement between the melts and/or methods. Either of these two approaches could be conducted with *MeltWin* or *meltR.A*. Unfortunately, both approaches are biased. To overcome this limitation, we provide an auto-baseline trimmer in *MeltR* that is consistent and rational, the "*BLTrimmer,"* which works on a *meltR.A* fit object (Figure 2A). In a typical analysis, the user fits the data with *meltR.A*, and can then optionally optimize the results using *BLTrimmer*, which uses three principles to optimize trimming:

1. Many trimmed baselines are generated and fit.
2. The best trims produce the most internally consistent folding thermodynamic parameter estimates.
3. Parameters from the best combinations are treated as an ensemble of equally feasible estimates.

Thus, *BLTrimmer* calculates parameters from a suite of differentially trimmed melts instead of a single trim. A description of *BLTrimmer* is provided in the supplemental methods (File S1). It has only one required argument, a fit object produced by *meltR.A* (Figure 2B.1), and prints an analysis of thermodynamic parameters and 95% confidence intervals (CI^{95}) that summarize an ensemble of baseline trims (Figure 2C). *BLTrimmer* also provides an analysis of all the tested baselines (Figure S2).

Other *BLTrimmer* arguments are set to a recommended value by default but are adjustable. The first adjustable argument is "Trim.method," which corresponds to the method used to generate baselines, either "floating" to allow different baseline lengths for each sample or "fixed" to use the same baseline length for each sample (Figure 2B.2). The "Assess.method" argument corresponds to the method used to evaluate trims for internal consistency (Figure 2B.3, Figure S2). It can be one of three integer values, "1" to optimize agreement between samples in method 1, "2" to optimize agreement between methods 1 and 2, or "3" (default) to optimize agreement between samples in method 1 and agreement between methods 1 and 2 (Figure S2). The "no.trim.range" argument defines the core melt region, by default between 5% and 85% unfolded (Figure 2B.4). The "quantile.threshold" argument corresponds to the fraction of trims to pass to the final ensemble (Figure 2B.5). By default, *BLTrimmer* will use the 10% of trims that produce the most consistent parameters.

The final adjustable arguments determine the diversity of baseline combinations and apply to the fixed or floating trim method. The “n.ranges” argument is the number of trims to generate per sample, and the “range.step” argument is the temperature step between trims (Figure 2B.6-7). By default, using the floating method, the *BLTrimmer* will produce six symmetrical baselines per sample, at lengths of 4, 8, 12, 16, 20, and 24 °C. The “n.combinations” argument allows the user to specify the number of random trims to test, by default 1000 (Figure 2B.8). Testing 1000 random, “floating,” combinations is enough to achieve convergence in 1 min (Figure S3).

MeltR accurately reproduces thermodynamic parameters

We first tested *MeltR* by fitting modeled data. We modeled a set of realistic melting curves at nine different C_ts for 96 two-state helices that include non-self-complementary, self-complementary, and monomolecular folding helices (see File S1 methods).[7,25] The modeled data were then fit with *meltR.A* and the *BLTrimmer*. Results for the three different methods—individual fits, $1/T_m$ versus ln C_t, and global fitting—were in good agreement with the known, input parameters (Figure S4A-D).

We next tested *MeltR* by fitting real, experimental datasets for 28 different helices, where each dataset contained melting curves of the same helix at 5-12 different C_ts (Table S1). We compiled 15 published datasets for monomolecular, self-structured-hairpin RNA,[26–28] 7 published datasets for self-complementary RNA,[19] and 6 new datasets collected on non-self-complementary RNA. The data were fit with *meltR.A* followed by the *BLTrimmer* (Table S1). For bimolecular helices, the three *MeltR* methods were in good agreement, with an average percent error between methods of 4.4%, 4.9%, 1.5%, and 0.6% for $\Delta H°$, $\Delta S°$, $\Delta G°_{37}$, and T_m, respectively. Likewise, for monomolecular helices, the two compatible *MeltR* methods, methods 1 and 3, were in good agreement, with an average percent error between methods of 3.1%, 3.1%, 4.7% (~0.14 kcal/mol), and 0.0% for $\Delta H°$, $\Delta S°$, $\Delta G°_{37}$, and T_m, respectively.

To obtain an independent measure of accuracy, we compared thermodynamic parameters calculated using *MeltR* to those using *MeltWin*. For the hairpin RNA and self-complementary duplexes, parameters calculated using *MeltWin* were from published sources[19,26–28] (Table S1), while for the

heteroduplexes, parameters calculated using *MeltWin* are presented for the first time (Table S1). Data from *MeltR* method 1 were in good agreement with those from *MeltWin* method 1 for bimolecular datasets, with an average %error in $\Delta H°$, $\Delta S°$, $\Delta G°_{37}$, and T_m between programs of 2.7%, 3.2%, 2.4%, and 2.1%, respectively (Figure 3A). Likewise, for method 2, data from *MeltR* were in good agreement with *MeltWin* for bimolecular datasets, with an average %error in $\Delta H°$, $\Delta S°$, $\Delta G°_{37}$, and T_m between programs of 2.8%, 3.3%, 2.2%, and 2.2%, respectively (Figure 3A). For monomolecular datasets, the average %errors were slightly larger, with an average error in $\Delta H°$, $\Delta S°$, $\Delta G°_{37}$, and T_m between programs of 4.1%, 4.0%, 6.3%, and 2.0% for method 1, respectively (Figure 3A). Even so, the absolute errors between programs for monomolecular and bimolecular structures were small, ~0.17 kcal/mol on average in terms of the $\Delta G°_{37}$. Interestingly, the unique *MeltR* method 3 of global fitting reproduces both *MeltWin* methods 1 and 2 for bimolecular and monomolecular datasets (Figure 3B-E).

We next tested *MeltR* parameters generated from fitting the real data for agreement with the predicted values from the consensus nearest neighbor model. Folding parameters for each of the 28 helices in the real datasets were calculated using published Watson-Crick nearest neighbor parameters.[7,19,26–28] Errors in $\Delta G°_{37}$ between the nearest neighbor model and *MeltR*/*MeltWin*-calculated values were 0.34-0.39 kcal/mol on average and smaller than 1 kcal/mol for all helices (Figure S4E). Lastly, there is no significant difference in errors in comparison to the consensus nearest neighbor model between any of the methods from *MeltR* and *MeltWin*, meaning that errors produced by *MeltR* are not statistically different from errors produced by *MeltWin* (Figure S4E). In conclusion, *MeltR* accurately determines folding parameters.

MeltR can test for two-state folding

Traditionally, helices have been tested for two-state folding by checking the agreement of $\Delta H°$ generated by methods 1 and 2. If the difference is <10-15%, the helix is considered two-state.[2,7,16,23] Unfortunately, the results of methods 1 and 2 are sensitive to baseline-trimming (Figure S2B and D), which leads to a question: Did user-imposed baseline trimming cause the appearance of two-state folding? *BLTrimmer*

provides a less biased approach based on a distribution analysis of Δ*H*° agreement derived from differentially trimmed baselines.

For example, we compared the distribution of Δ*H*° differences for a helix that is designed to exhibit two-state melting (Figure 4A) and another that is designed to exhibit three-state melting (Figure 4B). The three-state folding helix was designed to have GC- and AU-rich helical segments separated by an internal loop. The AU-rich end was expected to melt first, thus populating an intermediate, third state (Figure 4B). We then examined the distribution of the Δ*H*° agreement between methods 1 and 2 for 1000 baseline trims (Figure 4C). The distribution peaks at <5% for the first sequence, consistent with two-state folding, but at 15-18% for the second sequence, consistent with non-two-state folding. This distribution analysis provides a less biased metric of two-state folding compared to the analysis of a single baseline trim.

Discussion

We introduce *MeltR* and apply it to a series of simulated and real UV-detected absorbance melting data. We show that its output agrees with that of the legacy program *MeltWin*. Additionally, *MeltR* offers new features such as global fitting of data, auto-baseline trimming, and two-state tests. We found that *MeltR* was able to reproduce parameters using minimal user intervention (Figure 3A). Small differences between *MeltWin* and *MeltR* were likely caused by discrepancies between manual baseline trimming in *MeltWin* and automated baseline trimming in *MeltR*.

MeltR reports errors that represent the precision of parameters given the fit. The *meltR.A* function reports standard error values propagated from variance in the fit parameters (Figure 1D). The *BLTrimmer* reports CI^{95}s based on an analysis of an ensemble of baseline trims (Figure 2C). The advantage of CI^{95}s is that they do not assume a normal distribution of error and use an ensemble of data points to define a distribution. Although the errors reported by *MeltR* account for statistical errors

in the fits, they do not account for systematic errors in sample preparation, instrument calibration, extinction coefficients, and data analysis.[7] This explains why the global fitting method determines the $\Delta H°$ with a high precision of 0.2% at the *meltR.*A step (Figure 1D, SE values) but with a lower precision of 4% at the *BLTrimmer* step (Figure 2C, CI^{95}s). Specifically, global fitting with *meltR.A* is highly precise because it uses a single model to fit the dataset without accounting for systematic errors. This precision decreases using the *BLTrimmer* because it considers systematic errors in baseline trimming. The effect of incorporating this systematic error into the analysis is a sizable increase in uncertainty for the global fitting method. Furthermore, while the CI^{95}s provided by the *BLTrimmer* account for errors in baseline trimming, they do not account for additional systematic errors. Thus, the errors reported by *MeltR* in its entirety should be considered the minimum uncertainty in the parameters. To account for systematic errors in melt analysis, Turner and colleagues reported errors in thermodynamic parameters as a minimum flat percentage (12%, 13.5%, and 4% for $\Delta H°$, $\Delta S°$, and $\Delta G°_{37}$, respectively).[7]

Baseline trimming is a poorly understood and user-biased aspect of extracting parameters from melting curves.[24] The *BLTrimmer* aims to reduce user-bias and produce accurate parameters with minimal user specifications (Figure 3) by identifying an ensemble of optimum trims via agreement between different methods. A concern is that user-imposed trimming may cause a non-two-state dataset to appear two-state. In our experience, non-two-state data are identifiable by the distribution of $\Delta H°$ error for methods 1 and 2 (Figure 4), which can be automatically generated with *BLTrimmer* (Figure S2). We also recommend testing non-default *BLTrimmer* settings with a non-two-state dataset as a positive control. For user convenience, the three-state melting dataset presented in Figure 4 is preformatted and included in the *MeltR* package.

Ideally, users should design experiments so that melting curves exhibit T_ms between 35 and 65 °C, providing at least 30 °C to define lower and upper baselines in the standard experimental temperature range of 5 to 95 °C. However, this may not always be possible. In these cases, users may analyze data near this limit, if enough of the sigmoidal shape of the melting curve is defined. For example, our

analysis includes datasets collected on the self-complementary oligonucleotides 5′-UAUAUAUA-3′ and 5′-AGCCGGCU-3′, which have T_ms near 20 and 70 °C, respectively. Thus, lower baselines were not clearly defined for 5′-UAUAUAUA-3′, and upper baselines were not clearly defined for 5′-AGCCGGCU-3′. In both cases, *MeltR* was able to accurately determine thermodynamic parameters using automated baseline trimming in comparison to *MeltWin* (SI Table 1). For example, *MeltR* determined a $\Delta G°_{37}$ of -2.17 (-2.36 to -1.92) kcal/mol using method 2 on the low-melting, 5'-UAUAUAUA-3' dataset, within error of the *MeltWin*-determined $\Delta G°_{37}$ of -2.30 (±0.17) kcal/mol. This is within 0.25 and 0.12 kcal/mol for *MeltR* and *MeltWin,* respectively, of the -2.42 kcal/mol expectation for the consensus nearest neighbor model.

So far, we have intentionally limited our study to short (≤13 nt) RNA to avoid violation of the two-state assumption implicit to fitting data with *MeltR* and *MeltWin*. Even for short nucleic acids, great care must be taken for quantitative-analysis with two-state-models, to ensure that the data are consistent with a two-state melting transition, as we demonstrated in Figure 4. For long nucleic acids, a two-state melting transition is unlikely and quantitative analysis by fitting to a two-state model, in the absence of highly cooperative folding, is inappropriate. However, users may want to perform a semi-quantitative analysis of melting curves of long nucleic acids with *MeltR*. To demonstrate this application, we collected two datasets on long RNA, each dataset consisting of eight melting curves spanning a >50 fold concentration range for the cleaved-CPEB3 ribozyme (68 nt) and the *B. subtilis g*uanine riboswitch aptamer (73 nt). We first fit the datasets with *meltR.A* and plotted the first derivative analysis provided by *meltR.A*. The first derivative analysis indicates that CPEB3 melts in a single transition (Figure S5A) and is thus appropriate for semi-quantitative analysis by fitting to a two-state model. However, the first derivative analysis indicates that the guanine riboswitch aptamer melts in two distinct transitions (Figure S5B) and is thus inappropriate for semi-quantitative analysis by fitting to a two-state model. We then performed automated baseline trimming for the CPEB3 data with the *BLTrimmer* and compared the results to fits

with *MeltWin* in Table S2. The *MeltR* fits were within error of the *MeltWin* fits, indicating that *MeltR* is accurate in comparison to *MeltWin* for fitting long nucleic acids (Table S2).

In summary, *MeltR* provides the following advantages for fitting melting curves. First, it is as accurate as the comparable software *MeltWin*, while requiring less data processing time. Second, *MeltR* offers new features including global fitting, auto-baseline trimming, and two-state testing. Third, *MeltR* could provide a platform for facile and reproducible analysis of diverse thermal denaturation experiments. The signal need not be UV absorbance; CD- and fluorescence-detected melts are used to measure the stability of proteins and abiological foldamers and can be readily used in *MeltR*.[29–32] Fourth, *MeltR* is written in an accessible programming language. The source code can be modified and redistributed by the community without restrictions, providing a platform for others to expand and improve the code. Lastly, *MeltR* provides transparent analysis because it can be installed on any computer. The commands to analyze data can be distributed in a script that other researchers can reproduce. For example, we have provided all the data and fitting code for this paper at https://github.com/JPSieg/meltR.a.paper.

Materials and Methods

Data transparency and reproducibility

Data were analyzed in R (V4.2.1).[21] All data files, analysis code, and figures are available at https://github.com/JPSieg/meltR.a.paper. MeltR is available at https://github.com/JPSieg/MeltR and V1.0 is archived at https://zenodo.org/record/7501863.

Absorbance melting curves

For published data, unmodified data files were parsed into a tidy format. Long RNA were prepared as described previously.[22] For original experiments, short helices were ordered from IDT with RNase-free HPLC purification and dialyzed into buffer containing 1 M NaCl, 20 mM 3-(N-morpholino)propanesulfonic acid (pH 7.0), 0.01 mM ethylenediaminetetraacetic acid, and 0.001% (w/v) sodium dodecyl sulphate. Experiments were performed on an OLIS HP 8425 diode array

spectrophotometer from 5 to 95 °C at a ramp rate of 0.5 °C/min. Modeled melting curves were simulated to have random scatter and baseline slopes, as described in the File S1 methods. Data were fit with *MeltWin* as described previously[19] and with *MeltR*. Description of the methods used by *meltR.A* and the *BLTrimmer* is provided in the File S1 methods.

Supplemental Materials

File S1: Supplemental methods, figures, and tables

File S2: *meltR.A* help file

R-specific Definitions

Argument: The input a function needs to be executed.

Data frame: A table of data. Specifically, a list of vectors with each containing a different variable.

Fit object: An object produced by fitting data to a model containing useful data and statistics.

Function: Code that automates a task.

List: A type of data storage that can include vectors, data frames, and other lists.

Tidy formatted: A consistent data table format where each variable has its own column, each observation has its own row, and each value has its own cell.

Vector: Basic unit of data storage in R. A vector is created with the “c” function. For example, to create a vector for the “NucAcid” argument for *meltR.A*, one can type, “c(“RNA,” “AAAA,” “UUUU”).”

Acknowledgements

This work was supported by NIH Grants R35-GM127064 to PCB and R15-GM085699-04 to BMZ. We thank Dr. Susan Schroeder for the feedback.

Author contributions

JPS created the *MeltR* package under the supervision of PCB. SJA and BMZ tested and helped to improve *MeltR*. Data were collected or compiled and analyzed by JPS and SJA. All authors contributed to the writing.

Declaration of interests

All authors have no competing interests to declare.

References

(1) Puglisi, J. D.; Tinoco, I. Absorbance Melting Curves of RNA. In *Methods in Enzymology*; RNA Processing Part A: General Methods; Academic Press, 1989; Vol. 180, pp 304–325. https://doi.org/10.1016/0076-6879(89)80108-9.

(2) Schroeder, S. J.; Turner, D. H. Optical Melting Measurements of Nucleic Acid Thermodynamics. *Methods Enzymol.* **2009**, *468*, 371–387. https://doi.org/10.1016/S0076-6879(09)68017-4.

(3) Tinoco, I.; Uhlenbeck, O. C.; Levine, M. D. Estimation of Secondary Structure in Ribonucleic Acids. *Nature* **1971**, *230* (5293), 362–367. https://doi.org/10.1038/230362a0.

(4) Zuker, M.; Sankoff, D. RNA Secondary Structures and Their Prediction. *Bull. Math. Biol.* **1984**, *46* (4), 591–621. https://doi.org/10.1007/BF02459506.

(5) Sugimoto, N.; Nakano, S.; Katoh, M.; Matsumura, A.; Nakamuta, H.; Ohmichi, T.; Yoneyama, M.; Sasaki, M. Thermodynamic Parameters To Predict Stability of RNA/DNA Hybrid Duplexes. *Biochemistry* **1995**, *34* (35), 11211–11216. https://doi.org/10.1021/bi00035a029.

(6) SantaLucia, John; Allawi, H. T.; Seneviratne, P. A. Improved Nearest-Neighbor Parameters for Predicting DNA Duplex Stability. *Biochemistry* **1996**, *35* (11), 3555–3562. https://doi.org/10.1021/bi951907q.

(7) Xia, T.; SantaLucia, J.; Burkard, M. E.; Kierzek, R.; Schroeder, S. J.; Jiao, X.; Cox, C.; Turner, D. H. Thermodynamic Parameters for an Expanded Nearest-Neighbor Model for Formation of RNA Duplexes with Watson−Crick Base Pairs. *Biochemistry* **1998**, *37* (42), 14719–14735. https://doi.org/10.1021/bi9809425.

(8) Turner, D. H.; Mathews, D. H. NNDB: The Nearest Neighbor Parameter Database for Predicting Stability of Nucleic Acid Secondary Structure. *Nucleic Acids Res.* **2010**, *38*, D280–D282. https://doi.org/10.1093/nar/gkp892.

(9) Zuker, M. Mfold Web Server for Nucleic Acid Folding and Hybridization Prediction. *Nucleic Acids Res.* **2003**, *31* (13), 3406–3415. https://doi.org/10.1093/nar/gkg595.

(10) Markham, N. R.; Zuker, M. DINAMelt Web Server for Nucleic Acid Melting Prediction. *Nucleic Acids Res.* **2005**, *33* (suppl_2), W577–W581. https://doi.org/10.1093/nar/gki591.

(11) Gruber, A. R.; Lorenz, R.; Bernhart, S. H.; Neuböck, R.; Hofacker, I. L. The Vienna RNA Websuite. *Nucleic Acids Res.* **2008**, *36*, W70–W74. https://doi.org/10.1093/nar/gkn188.

(12) Reuter, J. S.; Mathews, D. H. RNAstructure: Software for RNA Secondary Structure Prediction and Analysis. *BMC Bioinformatics* **2010**, *11*, 129. https://doi.org/10.1186/1471-2105-11-129.

(13) Zadeh, J. N.; Steenberg, C. D.; Bois, J. S.; Wolfe, B. R.; Pierce, M. B.; Khan, A. R.; Dirks, R. M.; Pierce, N. A. NUPACK: Analysis and Design of Nucleic Acid Systems. *J. Comput. Chem.* **2011**, *32* (1), 170–173. https://doi.org/10.1002/jcc.21596.

(14) Ye, J.; Coulouris, G.; Zaretskaya, I.; Cutcutache, I.; Rozen, S.; Madden, T. L. Primer-BLAST: A Tool to Design Target-Specific Primers for Polymerase Chain Reaction. *BMC Bioinformatics* **2012**, *13* (1), 134. https://doi.org/10.1186/1471-2105-13-134.

(15) Peng, D.; Tarleton, R. 2015. EuPaGDT: A Web Tool Tailored to Design CRISPR Guide RNAs for Eukaryotic Pathogens. *Microb. Genomics 1* (4), e000033. https://doi.org/10.1099/mgen.0.000033.

(16) SantaLucia Jr., J.; Turner, D. H. Measuring the Thermodynamics of RNA Secondary Structure Formation. *Biopolymers* **1997**, *44* (3), 309–319. https://doi.org/10.1002/(SICI)1097-0282(1997)44:3<309::AID-BIP8>3.0.CO;2-Z.

(17) McDowell, J. A.; Turner, D. H. Investigation of the Structural Basis for Thermodynamic Stabilities of Tandem GU Mismatches: Solution Structure of (RGAG GU CUC) $_2$ by Two-Dimensional NMR and Simulated Annealing $^{\prime}$. *Biochemistry* **1996**, *35* (45), 14077–14089. https://doi.org/10.1021/bi9615710.

(18) Ghosh, S.; Takahashi, S.; Ohyama, T.; Endoh, T.; Tateishi-Karimata, H.; Sugimoto, N. Nearest-Neighbor Parameters for Predicting DNA Duplex Stability in Diverse Molecular Crowding Conditions. *Proc. Natl. Acad. Sci.* **2020**, *117* (25), 14194–14201. https://doi.org/10.1073/pnas.1920886117.

(19) Adams, M. S.; Znosko, B. M. Thermodynamic Characterization and Nearest Neighbor Parameters for RNA Duplexes under Molecular Crowding Conditions. *Nucleic Acids Res.* **2019**, *47* (7), 3658–3666. https://doi.org/10.1093/nar/gkz019.

(20) Hopfinger, M. C.; Kirkpatrick, C. C.; Znosko, B. M. Predictions and Analyses of RNA Nearest Neighbor Parameters for Modified Nucleotides. *Nucleic Acids Res.* **2020**, *48* (16), 8901–8913. https://doi.org/10.1093/nar/gkaa654.

(21) R Core Team. R: A Language and Environment for Statistical Computing. *R* **2022**.

(22) Sieg, J. P.; McKinley, L. N.; Huot, M. J.; Yennawar, N. H.; Bevilacqua, P. C. The Metabolome Weakens RNA Thermodynamic Stability and Strengthens RNA Chemical Stability. *Biochemistry* **2022**. https://doi.org/10.1021/acs.biochem.2c00488.

(23) Marky, L. A.; Breslauer, K. J. Calculating Thermodynamic Data for Transitions of Any Molecularity from Equilibrium Melting Curves. *Biopolymers* **1987**, *26* (9), 1601–1620. https://doi.org/10.1002/bip.360260911.

(24) Majikes, J. M.; Zwolak, M.; Liddle, J. A. Best Practice for Improved Accuracy: A Critical Reassessment of van't Hoff Analysis of Melt Curves. *Biophys. J.* **2022**, *121* (11), 1986–2001. https://doi.org/10.1016/j.bpj.2022.05.008.

(25) Antao, V. P.; Tinoco, I., Jr. Thermodynamic Parameters for Loop Formation in RNA and DNA Hairpin Tetraloops. *Nucleic Acids Res.* **1992**, *20* (4), 819–824. https://doi.org/10.1093/nar/20.4.819.

(26) Thulasi, P.; Pandya, L. K.; Znosko, B. M. Thermodynamic Characterization of RNA Triloops. *Biochemistry* **2010**, *49* (42), 9058–9062. https://doi.org/10.1021/bi101164s.

(27) Sheehy, J. P.; Davis, A. R.; Znosko, B. M. Thermodynamic Characterization of Naturally Occurring RNA Tetraloops. *RNA* **2010**, *16* (2), 417–429. https://doi.org/10.1261/rna.1773110.

(28) Saon, Md. S.; Znosko, B. M. Thermodynamic Characterization of Naturally Occurring RNA Pentaloops. *RNA* **2022**, *28* (6), 832–841. https://doi.org/10.1261/rna.078915.121.

(29) Becktel, W. J.; Schellman, J. A. Protein Stability Curves. *Biopolymers* **1987**, *26* (11), 1859–1877. https://doi.org/10.1002/bip.360261104.

(30) Kelly, S. M.; Jess, T. J.; Price, N. C. How to Study Proteins by Circular Dichroism. *Biochim. Biophys. Acta BBA - Proteins Proteomics* **2005**, *1751* (2), 119–139. https://doi.org/10.1016/j.bbapap.2005.06.005.

(31) Horne, W. S.; Johnson, L. M.; Ketas, T. J.; Klasse, P. J.; Lu, M.; Moore, J. P.; Gellman, S. H. Structural and Biological Mimicry of Protein Surface Recognition by α/β-Peptide Foldamers. *Proc. Natl. Acad. Sci.* **2009**, *106* (35), 14751–14756. https://doi.org/10.1073/pnas.0902663106.

(32) Alexandrov, A. I.; Mileni, M.; Chien, E. Y. T.; Hanson, M. A.; Stevens, R. C. Microscale Fluorescent Thermal Stability Assay for Membrane Proteins. *Structure* **2008**, *16* (3), 351–359. https://doi.org/10.1016/j.str.2008.02.004.

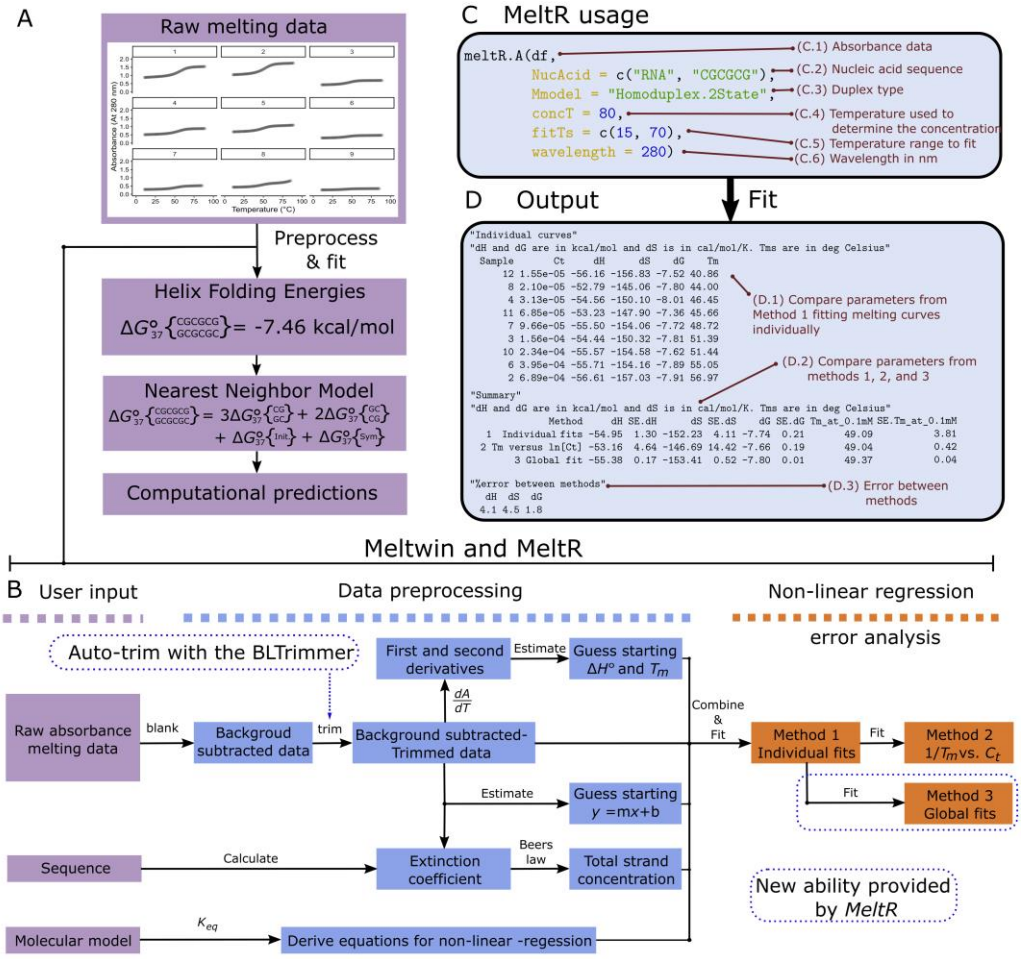

Figure 1 Determination of thermodynamic parameters from melting curves using *MeltR*. **(A)** Melting curves (top) underlie algorithms that predict nucleic acid structure and behavior. **(B)** Raw data-to-parameter conversion is automated with *MeltR* (shown here to demonstrate complexity; a full description is available in File S1). Flow chart shows three modules: input, preprocessing, and regression. Each arrow represents a step that must be undertaken, which converge at the end of data preprocessing. **(C)** Usage of *meltR.A* in an R script. (1) The data frame (df) containing the absorbance data. (2) A vector specifying the nucleic acid. (3) Duplex type: "Monomolecular.2State," "Heteroduplex.2State," or "Homoduplex.2State." (4) Temperature used to calculate the C_t. (5) Temperature range that is fit, used for manual baseline trimming. **(D)** Raw output of *meltR.A*. Note, the T_m versus ln[C_t] relationship is $1/T_m$ versus the ln C_t. All parameters are for folding rather than unfolding.

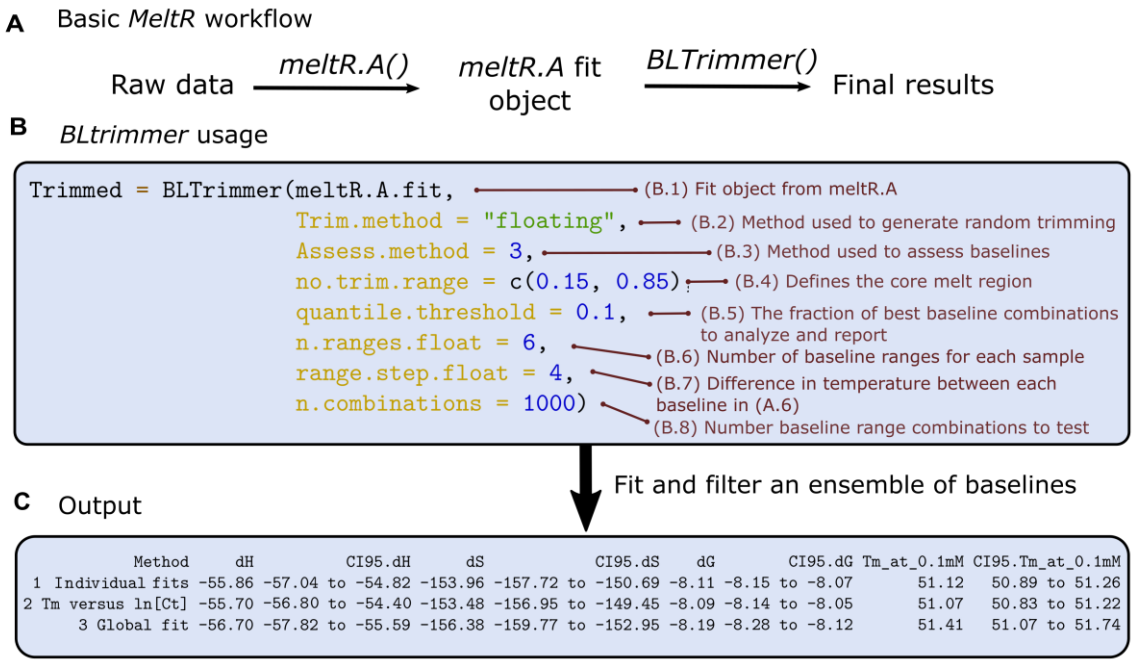

Figure 2 Auto-baseline trimming using the *BLTrimmer*. **(A)** Basic *MeltR* workflow. **(B)** *BLTrimmer* usage in an R script. (B.1) A *MeltR* fit object from *meltR.A*. (B.2) Method used to generate random baselines (either “fixed” or “floating”). (B.3) Method used to assess each baseline trim. (B.4) Defines the core melt region which is not trimmed. (B.5) Fraction of optimum baseline combinations to analyze in the final ensemble. (B.6) Number of trims the *BLTrimmer* will produce for each sample. (B.7) Temperature difference between each trim. (B.8) Number of baseline combinations to test. **(C)** Raw output of the *BLTrimmer*. All parameters are for folding rather than unfolding. CI^{95} represents 95% confidence intervals based on an ensemble of different baseline trims. Columns are right justified.

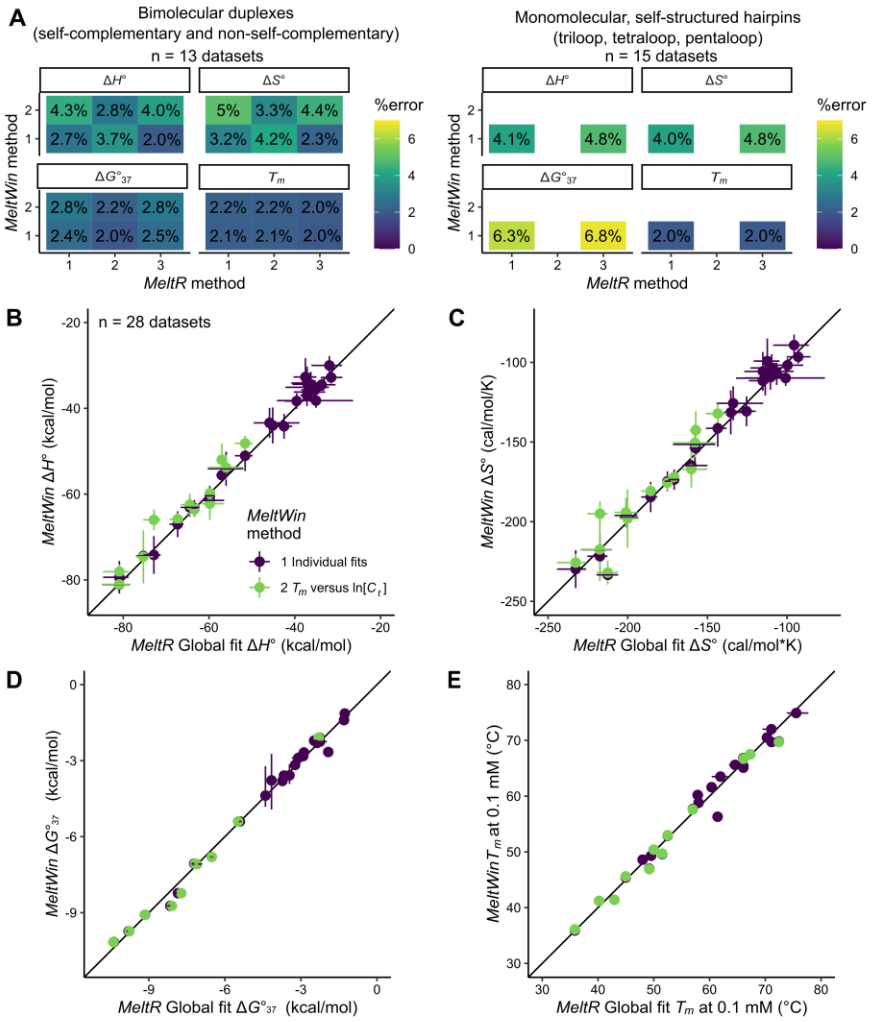

Figure 3 *MeltR* functions *meltR.A* and *BLTrimmer* reproduce folding parameters calculated using *MeltWin*. **(A)** Average %error between the three *MeltR* methods and the two *MeltWin* methods. Note that method 2 is incompatible with fitting data to a monomolecular, self-structured model because the T_m is not dependent on C_t. **(B-E)** Thermodynamic parameters from *MeltWin* method 1 (purple) and method 2 (green) versus parameters from *MeltR* method 3, global fitting. Vertical bars represent errors calculated by *MeltWin* and horizontal bars represent CI^{95}s calculated using the *BLTrimmer*.

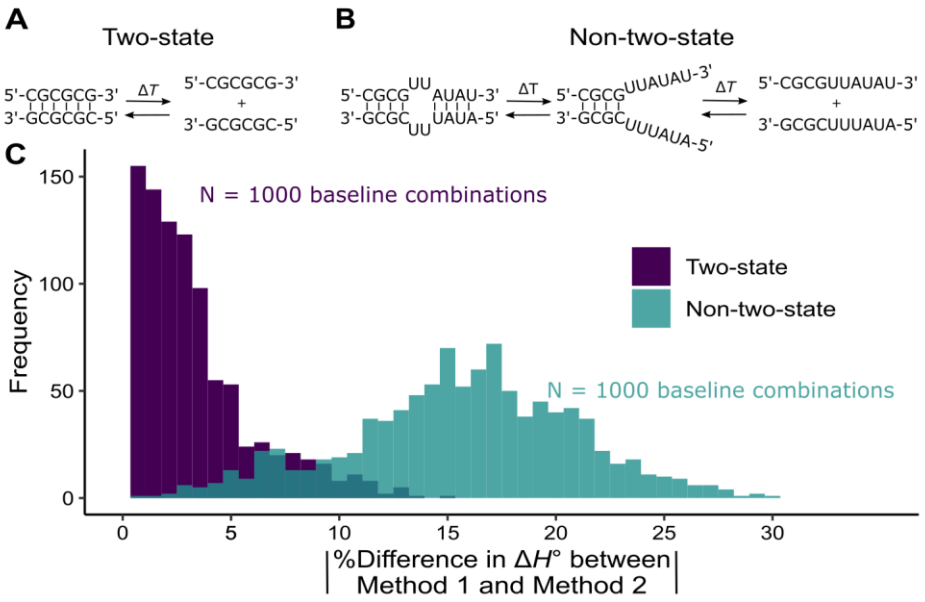

Figure 4 Two-state folding test using the *BLTrimmer* in *MeltR*. **(A)** Equilibrium for a two-state, self-complementary duplex RNA shown in the process of thermodenaturation. **(B)** Equilibrium for a non-two-state non-self-complementary, heteroduplex. **(C)** Distribution of the difference between the Δ*H*° from methods 1 and 2; absolute value taken. Purple and transparent teal histograms represent the two-state and non-two-state datasets, respectively. Each histogram represents the results of fitting 1000 randomly trimmed baseline combinations with methods 1 and 2.